Unsupervised Workflow Discovery in Industrial Environments

Fabian Nater[1] Helmut Grabner[1] Luc Van Gool[1,2]

[1]Computer Vision Laboratory
ETH Zurich
{fnater,grabner,vangool}@vision.ee.ethz.ch

[2]ESAT - PSI / IBBT
K.U. Leuven
luc.vangool@esat.kuleuven.be

Abstract

In this work, we present an approach for the automatic discovery of workflows in industrial environments. In such cluttered scenes, one faces many challenges, which limit the use of state-of-the-art object detection and tracking methods. Instead we propose a purely data-driven method which exploits the temporal structure of the workflow. Our robust technique is free of human intervention and does not need parameter tuning. We show results on two camera views of a working cell in a car assembly line. Workflows are extracted robustly, they match well across the camera views and they are conform with human annotation. Furthermore, we show a simple but efficient extension to analyze the image stream in real time. This assures a smooth running of the workflow and enables the notification of different types of unexpected scenarios.

1. Introduction

Surveillance tasks are nowadays increasingly augmented with vision systems and smart algorithms to extract information or detect precise (abnormal) events. In this work, we focus on the interpretation and analysis of industrial scenarios. Hereby, several challenges must be overcome, such as unfavorable working conditions with dust, sparks or vibrations, cluttered background, diverse moving objects or heavy occlusion of the workers. Additionally, the workers look very similar, as they often wear utility uniforms. In this context, one issue to monitor the smooth running of a workflow and detect any abnormal behavior. Deviations from the workflow may cause severe deterioration of the product quality or may raise safety or security hazards. Usually, the (normal) workflow has to be defined beforehand, which is done in an initial training phase with human intervention.

We propose a method to extract meaningful and interpretable workflows in an completely unsupervised manner. In order to overcome the involved challenges, we make use of clear assumptions that hold for industrial scenarios, such as the repeated structure of the workflow. To the best of our knowledge, we are the first to model workflows without any human intervention during the discovery process.

With our simple yet effective technique, we examine videos of an assembly line in a car manufacturing site. The extracted workflows turn out to be consistent across different camera views and well interpretable, also compared to independent human annotation. In addition, we analyze several hours of video data in real-time, which allows us to interpret the workflow, and reason on different abnormal situations. In fact, the obtained statistics can be used in order to optimize the workflow and enable a safe running of the monitored assembly process.

The paper is organized as follows. In the next section, we briefly outline relevant related work. The method is detailed in Sec. 3, where we first show the automatic extraction of a workflow and then its application for runtime analysis of unseen video. The experiments in Sec. 4 underline the use of our technique and paper is concluded in Sec. 5.

2. Related Work

The interpretation of a visual scene in order to extract useful information without human intervention is a popular field of research. For example, sophisticated techniques make use of specific constraints to discover object categories in images [5]. In video analysis, methods exist for various tasks, such as to extract trajectories of tracked objects [13, 7], to interpret motion in public places [6, 14], to learn human actions [11] or to model human activity patterns [9]. Most of these works rely on robustly detected and tracked agents in the scene. One step further, Zhou *et al*. have proposed techniques to segment [18] and discover [17] human actions or facial expressions. Their algorithms are unsupervised, however, they need to pre-define the desired number of states and require well defined features.

Relatively few work has been done for the analysis or automatic extraction of workflows. Recent medical applications use computer vision techniques to monitor surgical workflows [2, 12] in supervised settings. Due to the chal-

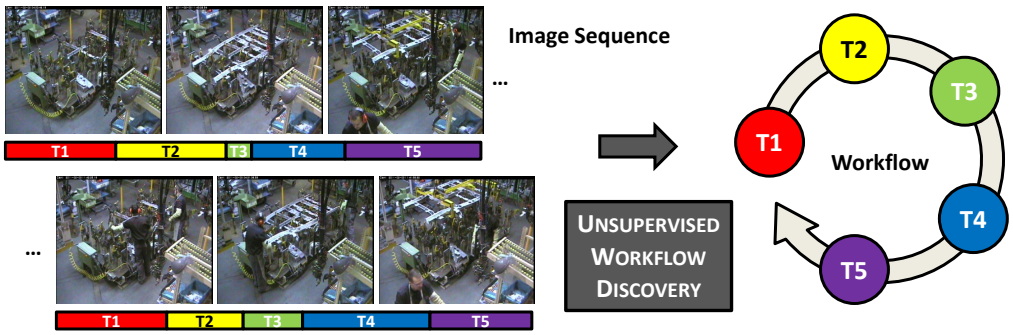


Figure 1. In industrial environments, assembly tasks typically have a repeated cyclic structure. They are called workflows and consist of several tasks. The number of tasks as well as the segmentation is unknown. The goal of this work is to extract the workflow in an unsupervised manner and provide a simple yet effective analysis of industrial activity.

lenging conditions in industrial environments, sophisticated image processing methods, such as the detection and tracking of objects or persons are hardly applicable. Approaches which build on these techniques are very likely to fail in practice. Hence, in the setting of industrial workflow monitoring, Veres *et al.* [15] proposed to use a holistic scene representation. The main drawback of all these approaches however is their need for a manually pre-defined workflow model and annotated tasks. They can only monitor, but not discover workflows.

3. Automatic Workflow Discovery

Given an image stream from a video camera, we aim to automatically discover the underlying workflow. No pre-segmentation of the image stream nor any other supervision is assumed to be available. Let us first define the following terms used:

Task: A task corresponds to a (physical) action, such as to pick up an object and place it somewhere.

Workflow: A workflow consists of a certain number of tasks and their transitions.

The goal of workflow discovery is to extract a number of N tasks T_n, with $n \in \{1, \dots N\}$ and N unknown, that represent the workflow observed in the scene.

3.1. Assumptions

As we aim for a widely applicable approach, we do not rely on explicitly modeling or recognizing humans, actions, or objects within the scene. Furthermore, we do not impose restrictions on the camera viewpoint. Yet, we have noticed some given factors that permit to set up assumptions concerning the nature of the workflow. They are described in the following.

Static camera: We assume that the workspace is monitored by a static camera.

Image sequence: We assume the image sequence to be temporally consistent, *i.e.*, neighboring image frames are correlated and are likely to share a common task label.

Cyclic workflow: We assume the workflow to have cyclic layout, *i.e.*, the tasks have always the same ordering and are repeated.

In other words, we are looking for a cyclic workflow observed by a video camera, as outlined in Fig. 1. These assumptions are usually satisfied in industrial assembly lines, where parts or goods are manufactured or assembled systematically in an identical and repetitive manner. In fact, it is essential to produce in regular working cycles in order to maximize output while reducing defects and wastes.

3.2. Workflow extraction

Our approach to the automatic discovery of a workflow makes use of the above assumptions and consists of (i) noise reduction for robust analysis, (ii) potential task spotting and (iii) temporal refinement. The individual steps are described in more detail in the following.

Noise reduction. The fact, that we are using a static camera, allows us to use the complete image and extract a holistic image representation. Given a sequence of images $\boldsymbol{x}_t \in \mathbb{R}^d$, we apply Principal Component Analysis (PCA) [1] on the zero-mean input feature vectors $\hat{\boldsymbol{x}}_t$. The data is projected onto its eigenvectors, and these projections, sorted with respect to the eigenvalues, span a new orthogonal space. In the first dimensions, maximal variance of the initial data is encoded, while dimensions with small eigenvalues most likely represent noise. We choose to select the $n_{PCA} \ll d$ first components in order to keep 80% of the total variance. $\boldsymbol{y}_t \in \mathbb{R}^{n_{PCA}}$ is the projection of $\hat{\boldsymbol{x}}_t$ onto these components.

Identification of potential tasks. It has been shown very recently that the temporal structure in image sequences provides a strong cue for learning representations [10]. Following this approach, we first learn an embedding using Slow Feature Analysis (SFA) that explores the temporal depen-

dencies in the data. Subsequently, we cluster the data in the obtained lowdimensional subspace.

Extraction of invariant signals. SFA [16] is a technique to automatically extract the invariant components in temporal signals. The output signal $\boldsymbol{z}_t$ of the SFA represents the slowest components in $\boldsymbol{y}_t$, *i.e.*, it minimizes the average temporal variation:

$$\min J_{SFA} = \min \mathbb{E}_t(\Delta \boldsymbol{z}_t), \text{ s.t. } \mathrm{Var}(\Delta \boldsymbol{z}_t) = 1, \quad (1)$$

where $\Delta \boldsymbol{z}_t = ||\boldsymbol{z}_t - \boldsymbol{z}_{t-1}||^2$. With the model $\boldsymbol{z}_t = \boldsymbol{w}^\mathsf{T} \boldsymbol{y}_t$, it can then be shown [16] that the solution to the generalized eigenvalue problem

$$\dot{\boldsymbol{D}}\boldsymbol{w} = \lambda \boldsymbol{D}\boldsymbol{w} \quad (2)$$

verifies the criterion of Eq. (1), where $\boldsymbol{D} = \mathbb{E}_t(\boldsymbol{y}_t \boldsymbol{y}_t^\mathsf{T})$ is the covariance matrix of the data and $\dot{\boldsymbol{D}} = \mathbb{E}_t\big((\boldsymbol{y}_t - \boldsymbol{y}_{t-1})(\boldsymbol{y}_t - \boldsymbol{y}_{t-1})^\mathsf{T}\big)$ is the covariance matrix of the temporal differences. The slowest varying features $\boldsymbol{z}_t$ are the projections of $\boldsymbol{y}_t$ onto the eigenvectors $\boldsymbol{w}$ associated to the smallest eigenvalues λ. We select n_{SFA} slowest dimensions which span the SFA subspace.

In fact, it has been shown for time series data that SFA yields the capacities of LDA if temporally adjacent samples are likely to belong to the same class and transitions are sparse [8]. This is verified in our setting from the assumption that a workflow consists of temporally consistent tasks.

Fig. 2 depicts the first four slow features over time for the industrial dataset used the experimental section of our paper (*c.f.* Sec. 4). The repeated workflow structure can be clearly observed.

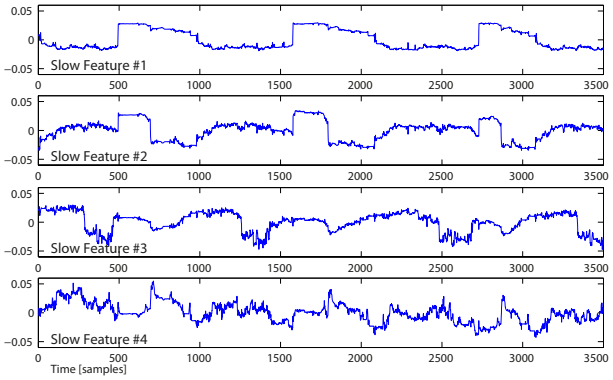


Figure 2. The four *slowest* features over time. The repetitive structure of the workflow appears in the first dimensions of the SFA subspace, whereas in higher dimensions, irregularities are encoded.

Please note, the slow features are extracted in an unsupervised manner. So, it is possible, that slow features also encode variations in the scene which do not belong to the workflow of interest. This might be due to other overlapping workflows (*e.g.*, bringing goods to the workplace), variations on a longer period of time (*e.g.*, illumination changes), or other background motion. However, in our experiments we did not observe such issues.[1]

Clustering. In the subspace of selected SFA components, the tasks appear as clusters of datapoints. We choose to apply mean shift clustering [3] because of its robustness and its capacity to discover nonlinear cluster structures. Furthermore, we do not need to manually fix the number of clusters to extract.

Following the SFA subspace properties, we choose the bandwidth of the mean shift kernel as the expected temporal variations $\mathbb{E}_t(\Delta \boldsymbol{z}_t)$. A two-dimensional SFA embedding and the obtained cluster centers in black are shown in Fig. 3.

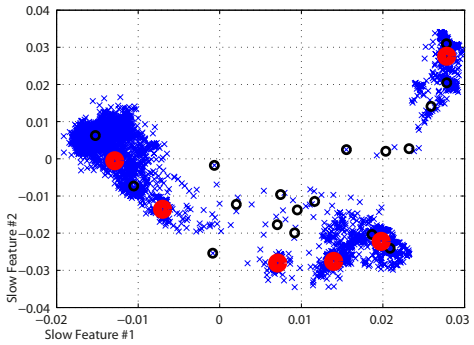


Figure 3. Mean shift clustering in two-dimensional SFA space, the initially detected 21 clusters (black) are refined to six final tasks (red) in the workflow.

Temporal refinement. Like the original data, the measurements in SFA space are always affected with noise. Based on the assumption of temporal consistency and of a cyclic workflow, we apply the following refinement steps:

Task duration: Tasks are required have a certain duration. Therefore, very short tasks which only consist of a few images are considered as noise (outliers) and are removed. In practice, we eliminate all clusters which are shorter than 5 seconds.

Cyclic workflow: By analyzing the task transitions, a cyclic workflow is enforced. Tasks are merged if they jitter or if they yield splits in the workflow.

In more detail, two tasks T_i and T_j are said to jitter and are merged if

$$P(T_i|T_j) > \Theta \wedge P(T_j|T_i) > \Theta, \quad (3)$$

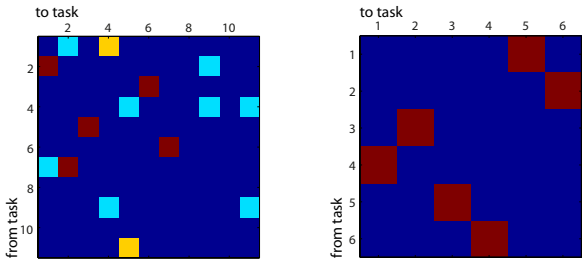
where $P(T_i|T_j)$ is the transition probability from T_j to T_i obtained from the clustered data. Θ is a small user defined threshold, we used $\Theta = 0.1$ for all experiments.

[1] Would such ambiguities appear, a similar method as in [4] could be employed. They use prior knowledge (breathing frequency) to select the desired SFA components for a medical application. So, we might use the desired length of a working cycle.

The assumption of a cyclic workflow implies a unique path, *i.e.*, from one task T_k only one dominant transition is allowed. Hence, we merge two tasks T_i and T_j if

$$P(T_i|T_k) > \Theta \wedge P(T_j|T_k) > \Theta. \quad (4)$$

To illustrate this procedure, exemplary task transition matrices before and after imposing the cyclic workflow layout are depicted in Fig. 4. The centers of the finally emerged tasks (clusters in the SFA subspace) are marked red in Fig. 3.



(a) (b)

Figure 4. Transition probabilities between the tasks: (a) after elimination of small (temporally short) tasks, (b) after imposing a cyclic workflow structure.

Model selection. In many subspace problems, it is unclear how to select the optimal number of latent dimensions. We propose to estimate this model complexity from the resulting number of tasks.

If a small number of dimensions is chosen, only few clusters emerge and the model of the working cycle might be overly simple with very general tasks. On the other hand, if we select more dimensions, many detailed states are identified, but they might degenerate and not fulfill the cyclic workflow assumption. Hence, these states are merged during refinement, which results again in a small number of final tasks. This said, we sweep over dimensionalities and choose the subspace such that the number of tasks in the workflow is maximized. This intuition is verified in Tab. 1, where the number of clusters (tasks) are indicated before and after refinement. In this case, we select the SFA subspace to be 2-dimensional ($n_{SFA} = 2$) and the discovered workflow comprehends six tasks.

SFA dimensionality	1	**2**	3	4	5	6
# of initial clusters	8	**22**	39	84	135	170
# of long tasks	6	**11**	13	13	12	12
# of final cyclic tasks	3	**6**	6	5	4	4

Table 1. Number of initially detected clusters, and discovered tasks as a function of the SFA subspace dimensionality. In this example, a two-dimensional representation is selected.

3.3. Analysis of unseen sequences

The analysis of new videos can provide statistical information on the tasks carried out, or it enables the detection of abnormalities in the observations.

Task classification. Our workflow discovery technique provides task labels to the initially unlabeled image sequence. With this information, any supervised classification method can be trained. In the following we show a very simple implementation.[2]

Training. Task classification is an multi-class classification problem. After PCA preprocessing, we opt to learn a representation of the labeled training data using Linear Discriminant Analysis (LDA) [1]. The LDA subspace, shown in Fig. 5, is discriminative and arranges the data in compact clusters $\mathcal{C}$.

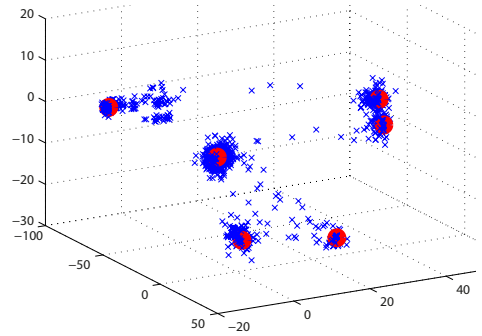


Figure 5. The six established tasks form compact clusters in the LDA space. At runtime, images are analyzed in this space

Runtime. At runtime, an image $\boldsymbol{x}$ is first projected into the LDA space to $\boldsymbol{x}'$. Then, the closest cluster center $\boldsymbol{c} \in \mathcal{C}$ determines the task label, *i.e.*,

$$T^*(\boldsymbol{x}') = \arg\min_{\boldsymbol{c}\in\mathcal{C}} ||\boldsymbol{x}' - \boldsymbol{c}||_2. \quad (5)$$

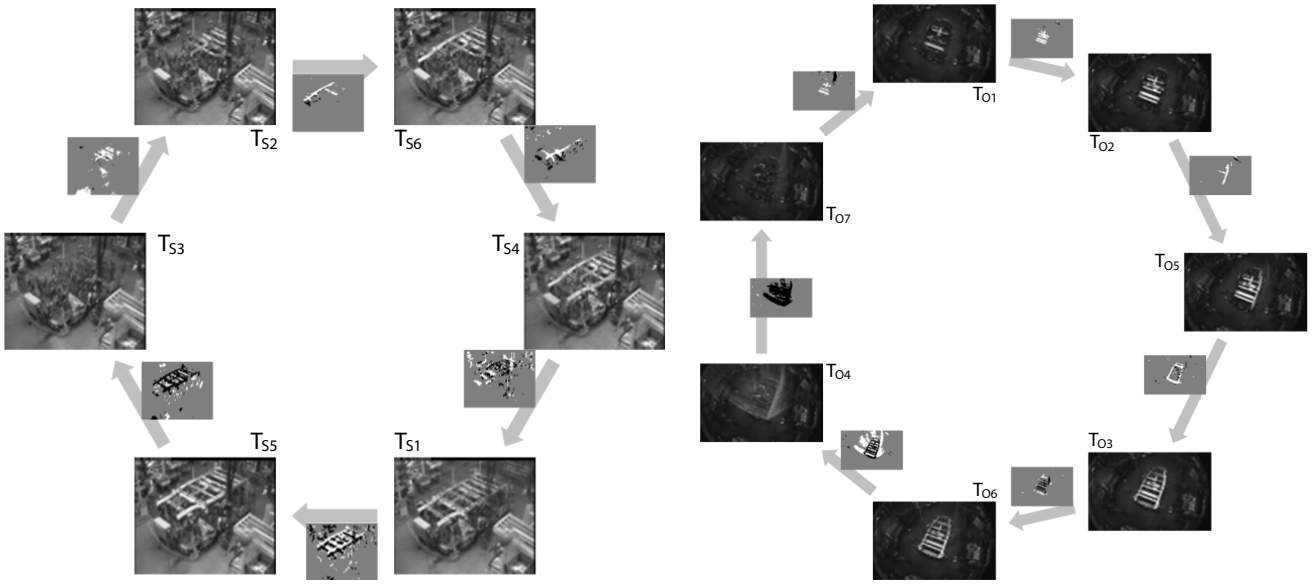
Anomaly detection. Three types of abnormalities can be detected with this simple model:

Appearance: Images which cannot be well assigned to any of the established clusters are considered as abnormal. This might be due to camera failures, large movements of the cameras or abnormal incidents in scene. To this end, we use the reconstruction error of the PCA model from the preprocessing step.

Sequence: The learned task sequence in the workflow should also be respected at runtime. If the task order changes, or a task is skipped, a problem can be signaled.

Timing: Each task is carried out for a certain duration. If the observed duration differs significantly from the trained one, a manufacturing issue might have occurred in this task.

[2]More sophisticated models, *e.g.*, using Hidden Markov Models, could certainly be learned, but do not fall within the focus of this paper.



(a) Side view (b) Overhead view

Figure 6. Automatically discovered cyclic workflows for the two camera views. The tasks are indicated with their mean images and the transitions are shown.

4. Experiments

Dataset. For our experiments, we use the data which was recorded in the SCOVIS project.[3] The data is recorded in a car manufacturing facility and the sequences show close views of an assembly area. Two camera views are provided, the first one monitors the working cell from the side and the second one is mounted overhead. The RGB-colored frames have a resolution of 704×576 pixels and are recorded at a framerate of $18 - 25$ fps. For the side view camera, recordings were made for approximately 1.5 working days.

Preprocessing. As input to our workflow analysis we convert the images to grayscale, downscale them by a factor of 8 (88×72) and finally reshape them to a 6336-dimensional feature vector. In all our experiments, we only analyze every 15^{th} frame.

4.1. Discovered workflows

We use the first hour of recordings for both camera views to apply our proposed automatic workflow discovery algorithm. The algorithm chooses in both cases a 2-dimensional SFA embedding. Details for the side view have been shown already as illustrative example in Sec. 3. In addition, the temporal sequence of tasks is shown in Fig. 7.[4]

Finally, the established workflows for the side view and the overhead view are depicted in Fig. 6 (a) and (b), respec-

[3] www.scovis.eu, 3^{rd} SCOVIS industrial dataset (shared upon our request and publicly available soon).

[4] The assignment of an index to a task is not necessarily in the order of the workflow due to the unsupervised clustering. However, if one likes, this can easily be done by switching the indices.

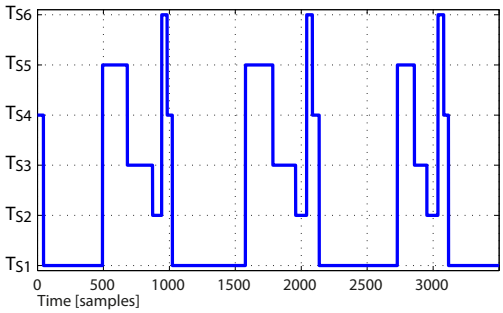


Figure 7. Sequence of matched clusters over time for an extract of the side view recordings in the discovery phase (also see the transition matrix in Fig. 4 (b)).

Manual task description	Side	Overhead
Two workers are putting a number of small spare parts (8 components) [...]	T_{S2}	T_{O1}, T_{O2}
Also they carry 2 big spare parts in the same table.	T_{S6}, T_{S4}	T_{O5}, T_{O3}
They are providing welding of the spare parts on the table construction.	T_{S1}	T_{O6}
One of them is manipulating and drives a yellow crane for taking the skeleton of the car in another plant.	T_{S5}	T_{O4}
This is the end of the workflow. The table plant is empty again and the workers start again [...].	T_{S3}	T_{O7}

Table 2. Comparison of our automatically detected workflow tasks with manual annotations.

tively. Tasks are represented by their mean images and are connected with directed arrows. The small images next to the arrows depict the average variations of the image intensities from one task to the next. Each task is numbered with the according task index.

For the side view, six tasks are discovered, whereas for the overhead view, seven tasks are found. It appears that the tasks correspond well between the two viewpoints. All tasks have its relative counterparts in the other view except for T_{O1} and T_{O2}, which are merged in the side view to T_{S2}. This is probably caused by the fact that the assembly of the different small spare parts is not very distinctive in the side view. Therefore, a single task is discovered, whereas in the overhead view, this placement is split into two separate tasks.

4.2. Comparison to manual annotation

A video extract of several workflow repetitions was viewed by an uninvolved person in order to describe the observed workflow in words. In Tab. 2 the annotated tasks are described and the reference is given for the two camera views. The automatically discovered tasks correspond very well to the tasks described by the human. Hence, the proposed technique is able to automatically extract tasks in a cyclic workflow, which are meaningful to human observers.

4.3. Runtime processing

For the side view camera, we apply the established workflow model on the recordings of the full day, *i.e.* approximately $40,000$ frames. Fig. 8 (a) depicts the tasks over time. Fig. 8 (b) and (c) show zooms of the long sequence, such that details become visible.

Statistics. In regular working cycles, the tasks are executed at regular speeds. Hence it is interesting to estimate the duration of each task from the data. A boxplot of timings for each task is shown in Fig. 9. T_{S4} and T_{S6} for example, are short and very regular. They correspond to the placement of the first and second long metallic bar, respectively. In the long task T_{S1} all the metallic parts are welded together. T_{S5} has a very variable timing. In this task, the assembled parts are delivered with a crane to another plant. Since it depends on the advancement of neighboring working cells, pauses occur in this task and its timing seems unpredictable.

Runtime. Since we only perform linear operations on the input features to project them into subspaces, the proposed analysis technique is very efficient. The actual algorithm runs at more than 25 fps on a standard PC using our MATLAB implementation.

4.4. Anomaly detection

During the runtime processing, several interesting cases are detected automatically:

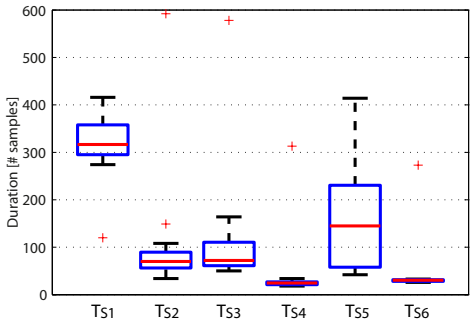


Figure 9. Boxplot for the duration of the extracted tasks. Please note that for better visibility, the y-axis is bounded, and not all outliers (crosses) are shown.

(a) Sparks at an abnormal location (b) Coffee break, ②

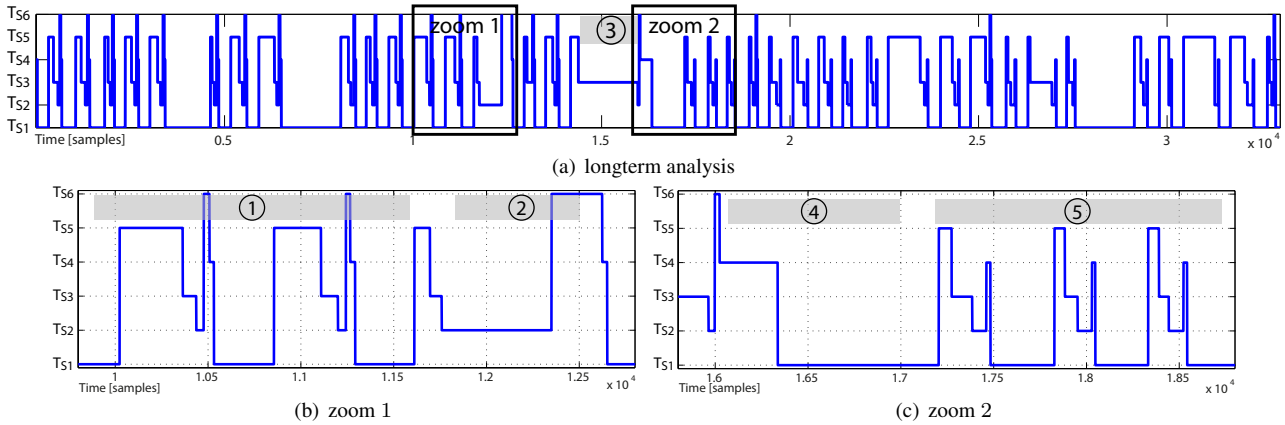
(c) Shift change, ③ (d) Cleaning, ④

Figure 10. Abnormal appearance (a) and abnormal timing (b)-(d) is detected automatically in the analyzed video.

Appearance. Fig. 10 (a) shows an exemplary abnormal event, which is detected because the image appearance does not apply well to any cluster. In this image, welding sparks appear at a very abnormal location, and we can suspect something abnormal going on here.

Timing. From the duration statistics in Fig. 9, abnormal timings of tasks can be identified. Three such cases are shown in Fig. 10 (b), (c) and (d). They correspond to the markers ②, ③ and ④ in the plots in Fig. 8, respectively and shows a work break at an unnatural instant within the working cycle, a worker shift change and a break for cleaning of the production space.

Sequence. During the analyzed work day, the sequential pattern of executed tasks changes. This appears from the comparison of markers ① and ⑤ in Fig. 8. A closer look is provided in Fig. 11, where it can be seen that two tasks are interchanged. During the workflow discovery process in the morning, the long metallic bar was first placed on



(a) longterm analysis

(b) zoom 1 (c) zoom 2

Figure 8. Analysis of unseen data with the learned workflow model. The matched tasks are plotted over time. One working day of video is analyzed and different anomalies are spotted. The markers refer to the descriptions in the text and Fig. 10 and 11.

the left, then on the right. Later on however, this learned cyclic structure of the workflow is no longer respected. The modification occurs right after the change of shift (marker ③). Apparently, the workers in the afternoon shift prefer to invert the order of the placement. This is not a critical issue here, but it could have been one. Nevertheless If the workflow discovery algorithm is run for a longer period of time, our approach will respect this task switch and will merge those two clusters. This would also be in line with the human interpretation of Tab. 2.

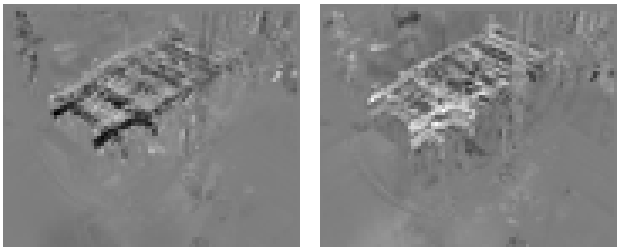
(a) morning shift ① (b) afternoon shift ⑤

Figure 11. Inverted tasks for morning (a) and afternoon shift (b). The placement order of the two long metallic bars is switched.

4.5. Discussion

As has been shown, our proposed algorithm is able to automatically extract meaningfull workflows. But what information in the images is really used? Since SFA is used in the discovery process, the structure of the embedding provides some information. The first two Eigenimages obtained by SFA projection are depicted in Fig. 12. As can be seen, the variance encodes the motion on the assembly table, which can be interpreted as the presence or absence of the parts. In contrast to many other methods which detect and track people, our approach does not focus on humans. The workers might even be considered as noise with respect to the entire workflow. Admittedly, they are somehow implicitly modeled, since they are necessary to bring the parts along.

We point out that this is not a general claim, but surly depends on the actual scenario. For another working cell, a person or other objects would well define the workflow. Due to the general structure of our data-driven algorithm, they would be picked up automatically in such cases. In summary, our algorithm chooses to model the workflow in the easiest possible way and respects the assumptions.



(a) first Eigenimage (b) second Eigenimage

Figure 12. Eigenimages of SFA (gray values corresponds to zeros, black to negative values and white to positive values). High variance is found on the assembly table, which can be interpreted as the presence or absence of parts.

5. Conclusion

In this work we presented a complete and automatic workflow discovery method. Exploiting the assumptions of temporal consistency and cyclic repeated patterns, we analyze the temporal structure of the image sequence. Without human interaction nor parameter tuning, cyclic workflows can be extracted robustly. The approach is tested on videos from two camera viewpoints and recorded within a challenging industrial environment. The discovered workflows match very well with human interpretations. We have shown that the discovered model can be used to obtain statistics, to optimize the workflow as well as for abnormality detection.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. 2, 4

[2] T. Blum, H. Feussner, and N. Navab. Modeling and Segmentation of Surgical Workflow from Laparoscopic Video. In *MICCAI*, 2010. 1

[3] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002. 3

[4] V. De Luca, H. Grabner, L. Petrusca, R. Salomir, G. Szekely, and C. Tanner. Keep breathing! Common motion helps multi-modal mapping. In *MICCAI*, 2011. 3

[5] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *Proc. CVPR*, 2008. 1

[6] T. Hospedales, S. Gong, and T. Xiang. A Markov Clustering Topic Model for mining behaviour in video. In *Proc. ICCV*, 2009. 1

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, F.-T. Tan, and S. Maybank. A System for Learning Statistical Motion Patterns. *PAMI*, 28(9):1450–1464, 2006. 1

[8] S. Klampfl and W. Maass. Replacing supervised classification learning by Slow Feature Analysis in spiking neural networks. In *NIPS*, 2009. 3

[9] F. Nater, H. Grabner, and L. Van Gool. Exploiting Simple Hierarchies for Unsupervised Human Behavior Analysis. In *Proc. CVPR*, 2010. 1

[10] F. Nater, H. Grabner, and L. Van Gool. Temporal relations in videos for unsupervised activity analysis. In *Proc. BMVC*, 2011. 2

[11] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, 79(3):299–318, 2008. 1

[12] N. Padoy, D. Mateus, D. Weinland, M.-O. Berger, and N. Navab. Workflow Monitoring based on 3D Motion Features. In *ICCV Workshop on Video-oriented Object and Event Classification*, 2009. 1

[13] C. Stauffer and W. E. L. Grimson. Learning Patterns of Activity Using Real-Time Tracking. *PAMI*, 22(8):747–757, 2000. 1

[14] M. Turek, A. Hoogs, and R. Collins. Unsupervised Learning of Functional Categories in Video Scenes. In *Proc. ECCV*. 2010. 1

[15] G. Veres, H. Grabner, L. Middleton, and L. Van Gool. Automatic Workflow Monitoring in Industrial Environments. In *Proc. ACCV*, 2010. 2

[16] L. Wiskott and T. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, 2002. 3

[17] F. Zhou, F. De la Torre, and J. F. Cohn. Unsupervised Discovery of Facial Events. In *Proc. CVPR*, 2010. 1

[18] F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conference on Automatic Face and Gestures Recognition*, 2008. 1