Workflow Monitoring based on 3D Motion Features

N. Padoy[1,3] D. Mateus[1] D. Weinland[2] M-O. Berger[3] N. Navab[1]

[1]TUM, Munich, Germany
{padoy,mateus,navab}@cs.tum.edu

[2]EPFL, Lausanne, Switzerland
daniel.weinland@epfl.ch

[3]INRIA, Nancy, France
{padoy,berger}@loria.fr

Abstract

Activity recognition has primarily addressed the identification of either actions or well-defined interactions among objects in a scene. In this work, we extend the scope to the study of `workflow` *monitoring. In a workflow, ordered groups of activities (phases) with different durations take place in a constrained environment and create temporal patterns across the workflow instances. We address the problem of recognizing phases, based on exemplary recordings. We propose to use Workflow-HMMs, a form of HMMs augmented with phase probability variables that model the complete workflow process. This model takes into account the full temporal context which improves on-line recognition of the phases, especially in case of partial labeling. Targeted applications are workflow monitoring in hospitals and factories, where common action recognition approaches are difficult to apply. To avoid interfering with the normal workflow, we capture the activity of a room with a multiple-camera system. Additionally, we propose to rely on real-time low-level features (3D motion flow) to maintain a generic approach. We demonstrate our methods on sequences from medical procedures performed in a mock-up operating room. The sequences follow a complex workflow, containing various alternatives.*

1. Introduction

In recent years, the analysis of human activities from videos has drawn an increasing interest in the Computer Vision community. The trend is motivated, first, by the wide variety of applications (medical monitoring, surveillance, human-machine interaction, etc.) concerned with understanding and modeling human behaviors, and second, by the fact that videos provide valuable information about the environment and are easy to obtain. Previous works in the domain have addressed different problems such as recognition of human actions [1]; activity and anomaly detection in public environments [6], modeling and identification of the primitive actions composing activities [11]; automatic discovery of the activities [7] and recognition of events in groups, such as in meetings [14].

A common underlying feature in the above-cited works is their ability to recognize *isolated* actions or activities (*e.g*. pick up, wave, fight, etc). In this way, they comply with most long-term applications, where relevant actions need to be detected but periods of inactivity and uninteresting actions should be discarded. Instead, we address in this paper the activity recognition problem in the context of a *workflow*. In this case, activities follow a well-defined structure in a long period of time and can be semantically grouped in relevant *phases*. The major characteristics of the phase recognition problem are the temporal dependencies between phases and their highly varying durations.

To perform real-time recognition of phases, we propose to use Workflow Hidden Markov Models (WHMMs). These are hierarchical HMMs that model the complete workflow process including its alternatives. WHMMs contain additional phase probability variables that keep track of semantic information about the phases in the model. Domain knowledge is provided in the form of a few labeled videos recorded from the workflow, indicating the interesting phases. Initialization of WHMMs is performed automatically from the labeled sequences.

Exemplary environments where workflow analysis is of interest are production lines in factories or hospital Operating Rooms (ORs). We are particularly interested in the medical application. Recent discussions about *the OR of the Future* [4] foresee ORs will become increasingly high-tech and specialized for a few kinds of surgeries. Real-time recognition of high-level phases in the OR (see Fig. 1 and Fig. 2 for an illustration of a surgical workflow) can improve synchronization between the ORs, provide a context aware support to the surgical staff and permit automatic and objective documentation.

The workflow in the OR comprises multiple, precise and complex activities usually involving the interaction of several people and objects. Furthermore, when observing the scene with a camera, colors on clothes and tissues are similar and multiple occlusions occur as the personnel principally work around a small area around the patient table. For these reasons, tracking and recognition of human actions in this specific environment are impractical and furthermore, not absolutely necessary for phase recognition. Due to specialization and sanitary conditions, high-level phases containing these activities resemble from one surgery to another and follow a regular workflow. As we will show later, a global model of the scene is sufficient for recognition since the whole activity focus is on the patient. To capture this activity coherence, we propose to use a multiple-camera system and a real-time reconstruction algorithm. This choice guarantees that the system will not interfere with the normal behavior of the medical staff in the OR and permits the usage of generic low-level features to characterize the phases.

The main contributions in this paper are as follows. We introduce and formalize the problem of recognizing *phases* in a *workflow* with alternatives. We propose WHMMs, a form of HMMs using phase probabilities to deal with semantic loss during Expectation Maximization (EM) training. Parameters of the model are initialized in a simple but effective way from annotated sequences. We show the benefits of complete workflow modeling and phase probability variables in *on-line* and *off-line* recognition results, also using partially labeled data. Finally, we propose to use 3D motion-flow for recognition in complex environments.

The remainder of the paper is organized as follows. Related work is discussed in Sec. 2. In Sec. 3, we give a formal definition of workflow and phases before stating the phase-recognition problem. In Sec. 4, we describe our proposed solution based on WHMMs and phase probabilities. Sec. 5 details the 3D-flow features. Experimental results are reported in Sec. 6. The conclusion in Sec. 7 includes some discussion on the method and future work.

2. Related Work

Pinhanez and Bobick [15] were among the first to analyze complex flows of actions. In their seminal work, framing in a TV studio was proposed to be done automatically, by triggering cameras based on the detection of events specified in a script. Unfortunately, their vision system was not able to detect the events and the authors were forced to generate them manually. Although progress has been done ever since [15], using vision systems for recognition in complex environments remains difficult. The problem is usually simplified by limiting the number of actors or that of actions, using a distinctive background, restricting the activity area or discriminating the actions by their spatial location.

Koile *et al*. [9] introduced the concept of *activity zones*, regions in an environment that are linked to specific activities. Likewise, Nguyen *et al*. [13] represent the room as a collection of cells. In both cases a tracking system is used to determine the presence of a person in a zone or the occupation of a cell. The goal of [13] is to recognize behaviors that

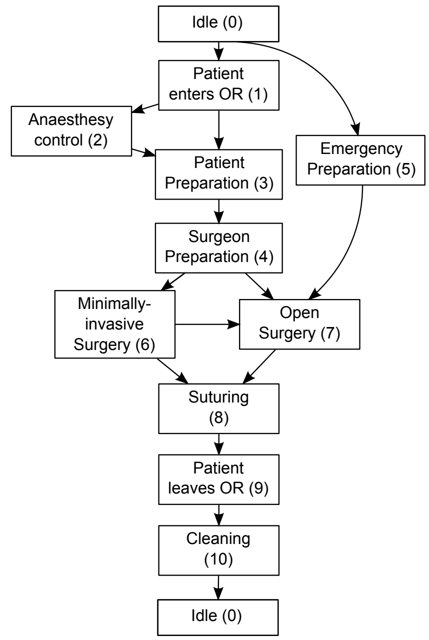

Figure 1. Scenario describing the operating room workflow with the phase labels in parentheses.

differ in the occupied cells and in the sequence of their occupation. The behaviors/primitives hierarchy used in [13] can be assimilated to our definition of workflow/phases, since they both impose temporal constraints. In practice, we deal with a larger time-scale, which makes our problem more complex. Additionally, our signals contain less semantics about the underlying actions and occupation alone does not provide enough information in a crowded scenario.

The use of model constraints to recognize complex events has been suggested before in works like [20, 11, 18, 17]. Xiang *et al*. [20] address structure learning in HMMs to obtain temporal dependencies between a few high-level events for video segmentation. An HMM models the simultaneous output of event-classifiers to filter their wrong detections. Moore and Essa [11] use stochastic context-free grammars to recognize separable multi-tasked activities in a card game from video. Production rules were manually defined to describe all the relations between the tracked events. Vu *et al*. [18] uses a symbolic approach to recognize complex activities in surveillance applications. For each activity, a formal scenario is provided by hand, including actors, objects and their spatio-temporal dependencies. Oliver *et al*. [14] propose layered HMMs for event recognition in meetings. The HMMs run in parallel at different levels of data granularity which permit event classification using multi-modal information. Each of the HMMs has to be provided with its respective data for training. Shi *et al*. [17] propose propagation networks (P-nets) to model and detect from video the primitive actions of a task performed by a tracked person. P-nets explicitly model parallel streams of

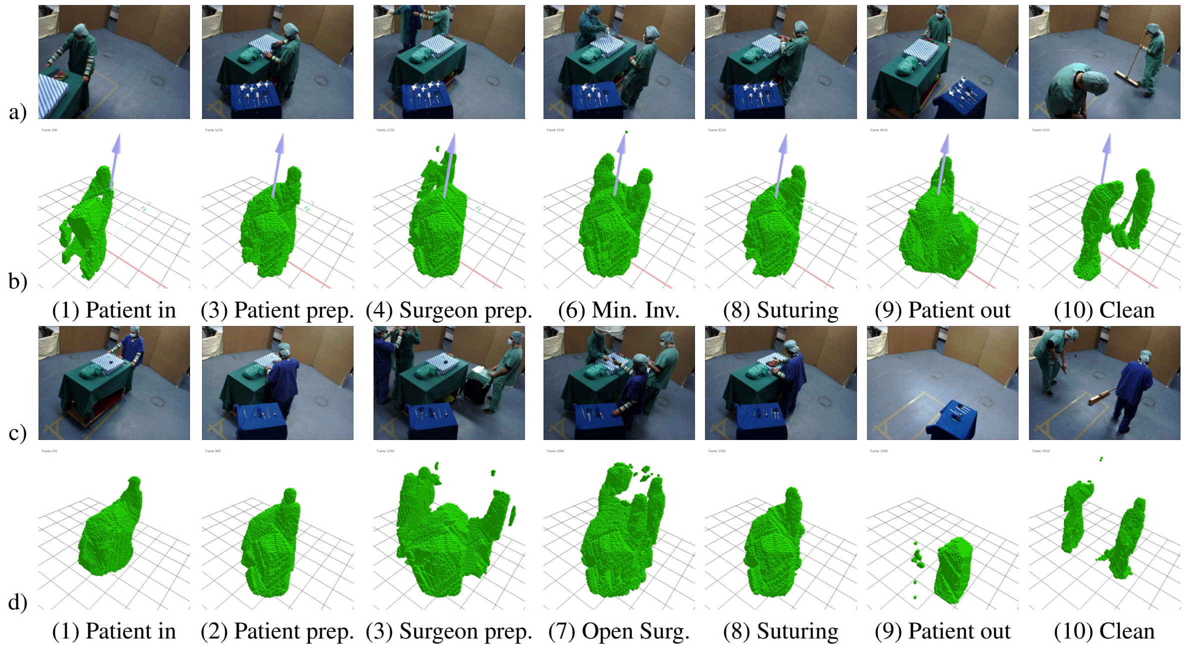

Figure 2. Two instances of the workflow. Images from one view and associated 3-D real-time reconstruction. Rows a,b) : a minimally-invasive surgery. Rows c,d) : an open-surgery.

events and are used for classification. The detailed topology is handcrafted and trained from partially annotated data.

Even though the aforementioned works share our initiative of imposing constraints on the flow of events to design a better model and improve recognition results, their main focus is on detecting single instances of actions. Our goal is to recognize phases containing multiple actions within a long-term, high-level workflow with alternatives. In brief, our work is different in that it considers either a higher semantical level (the phase) or an environment with more constrained temporal repetitivity.

Concerning the computation of features to observe the scene from video data, the works cited above mainly rely on detection and tracking of the persons and objects. Failure of one of these components hinders the recognition of the events. Actually, only few works address recognition directly using 3D reconstruction data. In [10, 19], actions performed by a single human, such as kicking or punching, are learned from 3D voxels. View-invariant action recognition is then performed from 2D shapes. 2D optical flow was recently used for action recognition by Efros *et al*. [5]. To our knowledge, no previous work has used 3D flow for the analysis of complex workflows.

3. Problem Statement

We begin by defining the usage of certain terms in order to formally state the problem. An *action* is the fundamental *element* in the semantic interpretation of a scene; it consists *e.g*. in picking up, handing, waving, etc. When several of these actions are considered together, they form *activities* or *behaviors* [3]. With increasing complexity and timescale, we define a *phase* as a semantically meaningful group of activities occurring somewhere inside a temporal sequence of actions, *i.e*. a "step" in a process. Phases occur *repetitively* across different sequences; the order in which they appear matters. Phases can have huge differences in durations, even across recordings. Notice that the semantic relevance which leads to considering a group of activities as a phase, is determined on the basis of domain knowledge. Finally, a set of phases along with their temporal relations are named a *workflow*.

3.1. The Phase Recognition Problem

The goal of phase recognition is to determine which phases occur and when, while observing an instance of a given workflow described by a temporal sequence of T observations $\{\mathbb{O}_{1:T}\} \subset \mathcal{O}$. If we give to each phase a label p from a set of labels $\mathcal{L}$, the problem reduces to finding a label assignment $\mathcal{P}$ for each time-step: $\mathcal{P} : \{1, \ldots, T\} \rightarrow \mathcal{L}$.

The challenge is to construct a model that explains the workflow sufficiently well to allow the estimation of $\mathcal{P}$ from the observations. In the next section, we explain how to build such a model from the a-priori knowledge contained in a few labeled exemplary sequences of the workflow.

Once the model is built, we address two forms of the recognition problem: *off-line* when the complete observations $\mathbb{O}_{1:T}$ is available; or *on-line* on the basis of partial observations $\mathbb{O}_{1:t}$ up to time $t \leq T$.

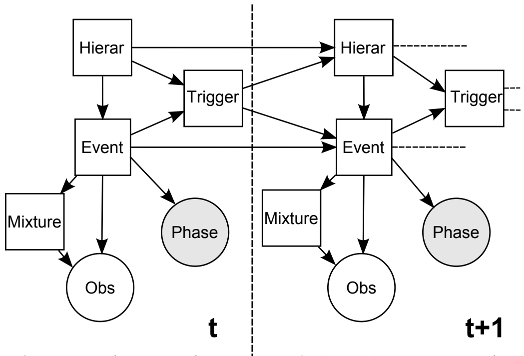

Figure 3. Dynamic Bayesian Network (DBN) representation of the WHMMs: two-levels hierarchy with a phase variable.

4. Proposed Method for Workflow Modeling

As mentioned previously, given an observation sequence our aim is to recognize (on-line or off-line) at each time step the current phase of the workflow. We therefore learn a statistical model of all phases and their dependencies. The model, which we call Workflow-HMM (WHMM), represents a hierarchical HMM that we augment with additional *phase-probability variables*. Effectively, the model has a two-layer hierarchy: nodes on the top layer represent dependencies between distinct phases, while nodes on the bottom layer represent dependencies within individual phases. The additional phase variables link those *latent* nodes to semantic meaningful phase labels, and hence allow us to infer the phase at each time step. As we will detail in section 4.2, those variables are in particular necessary to keep track of a semantic shift [17] that can occur between latent states and observations during a global EM training.

The detailed structure of our WHMM is illustrated using the convenient graphical description of Dynamic Bayesian Networks[12] in Fig. 3. In this formalism, the random variables *Hierar*, *Event* and *Trigger* enforce the two-level hierarchy, which models the phases and their dependencies. *Mixture* and *Obs* represent the observation distributions occurring within the process. Finally, *Phase* models the probabilities of being in a phase knowing the current *Event*. Note that different architectures could also be plugged into our framework. Our main objective is to show the general usefulness of phase probabilities for workflow monitoring with a widely-used kind of HMM.

We construct the WHMM using labeled and unlabeled observation sequences of the workflow. The construction involves the generation of the top-level topology, the initialization of the parameters in the lower hierarchy (Sec. 4.1), and the overall training using EM (Sec. 4.2) to refine all probabilities. For initialization, we only use labeled data, and hence do not require structural learning. This is in particular of advantage when we only have a small set of training sequences available. Moreover, by using the labeling we can directly derive a meaningful high level topology. Consequently, learning of the model can be simplified by using a conventional HMM with a single state variable x, which encodes all possible combinations of the Markov states of the original model. The topology shown in Fig. 3 is nevertheless preserved by enforcing its structure on the transition matrix of x, *i.e.* by setting non-possible transitions to zero.

Formally, the resulting HMM is a sextuplet $\lambda = (N, A, \mathcal{O}, B, \pi, \phi)$ where N is the number of states $\{x_i : 1 \leq i \leq N\}$ in the model, A the transition probability matrix between the states, modeling the topology, and $\mathcal{O}$ the space of observations. B is the observation model, indicating for any observation $\mathbf{o} \in \mathcal{O}$ and state x the probability $B_x(\mathbf{o}) = P(\mathbf{o} \mid x)$ that $\mathbf{o}$ can be observed by x. π is a probability distribution over the initial states. ϕ denotes the *phase probability variables*, indicating the probability $\phi_x(p) = P(p \mid x)$ of *each* state x to belong to a phase $p \in \mathcal{L}$. Relating this definition to Fig. 3, A models the dependencies of random variables *Hierar*, *Event* and *Trigger*; $\mathcal{O}$ of *Mixture* and *Obs*; ϕ of *Phase* and *Event*.

4.1. Initialization of Model Parameters

The initialization consists in two steps: 1) generating the top-level topology by enforcing the temporal constraints between the phases using a set of labeled sequences $\{\mathbb{O}^l\}$.
2) initializing the bottom level from the labeled information.

As each labeled sequence provides the temporal relationships between its labels, a directed graph can be deterministically derived from the data, modeling the temporal relationships between the phases, *i.e.* the workflow (as illustrated in Fig. 4). For the sake of recognition, inter-phases, namely the chunks of observations between two consecutive labels, are also used to build nodes of the graph. Inter-phases model background and intermediary activities, which are also repetitive across instances of the workflow.

Fig. 4 also shows that the resulting initialization is obtained by replacing each node of this graph with a sub-HMM modeling the corresponding group of events. Each sub-HMM is first initialized using the sub-sequences of data corresponding to its label. The number of states is defined based on the average-length of the sub-sequences and the transition probabilities are set so that the expected duration corresponds to the duration of the phase. We use a left-right skip-one-ahead transition model with mixtures of gaussians to model the observations. The mixtures are initialized using k-means from temporal splits of the training sub-sequences. Each sub-HMM is finally trained independently with EM on the sub-sequences.

4.2. Model Refinement and Phase Probability Variables

After learning the individual sub-HMMs (white nodes Fig. 4), we apply a global EM training to the complete model using all available training sequences (labeled and

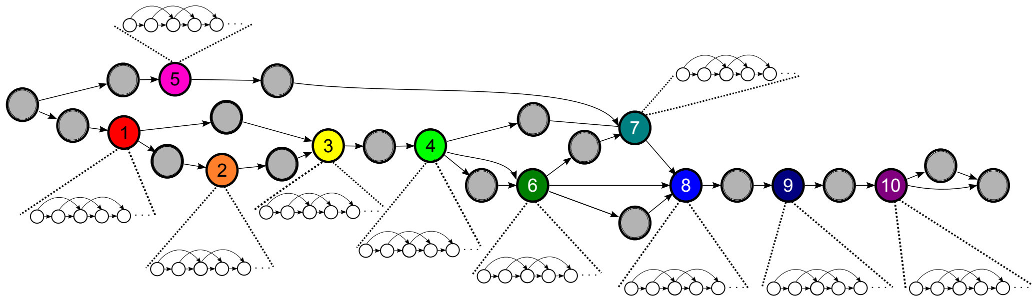

Figure 4. Graph representing the temporal relationships between the phases of workflow in Fig. 1, as extracted from annotated sequences. Colored nodes stand for labeled phases and gray nodes for inter-phases. Bottom levels of gray nodes are not displayed in this figure.

non-labeled), to refine the dependencies between different phases and their overall structure. At this point, the additional phase probabilities $\phi_x(p) = P(p \mid x)$ come to importance, as they allow to keep track of a possible shift that can occur between state variables and the associated labels.

Initially, after training individual sub-HMMs, each hidden state clusters only observations of a single phase, and consequently phase probability variables ϕ_x are simply a binary indicator of the respective phase. When applying global EM, however, observations can shift to neighboring states, and consequently hidden states might no longer exclusively represent single phases. Phase probabilities are hence used to track this shift.

Different proposals have been made to deal with such a shift in the semantic meaning of the hidden states. For instance Shi et al. [17] proposed for learning of their *propagation networks* the use of a labeled anchor sequence during EM to prevent alteration of semantic information. In that work, sub-actions are represented only through single nodes and the model topology is provided manually. When the topology is derived from the data and the modeling involves longer and more complex sub-phases represented by several nodes, constraining the EM algorithm is less natural. Instead, we propose a general approach that keeps track of the phases precisely by using the *phase probability variables*. This provides moreover a convenient formulation during on-line recognition in the WHMM (see Sec. 4.3) and is potentially more powerful by permitting the addition of new labels without performing another EM training.

We compute ϕ_x a-posteriori using a set of labeled sequences $\{\mathbb{O}^l\}$, in a way similar to a single update of model parameters during EM [16]. Using $\gamma_t^l(x) = P(X_t = x | \mathbb{O}^l)$, indicating the probability to be in state x at time t while knowing the sequence $\mathbb{O}^l$, we count the number of visits that a label makes to each state:

$$\phi_x^l(p) = \frac{\sum_{\{t, phase(\mathbb{O}_t^l)=p\}} \gamma_t^l(x)}{\sum_t \gamma_t^l(x)}. \quad (1)$$

Finally, ϕ_x is obtained by taking all available sequences into account.

4.3. Recognition

According to the available observation data at the moment where the recognition is performed, the computation of $\mathcal{P}$ can be done off-line or on-line, as follows:

Off-line: The Viterbi algorithm [16] is used to find the most likely path through the topology of the WHMM, *i.e.* $path : \{1, \ldots, T\} \rightarrow \{(x_i)_{1 \leq i \leq N}\}$. Using $path$, the sequence is labeled by:

$$\mathcal{P}(t) = \operatorname*{argmax}_p \phi_{path(t)}(p) \quad . \quad (2)$$

On-line: In the WHMM model, the forward probabilities [16] permit to compute:

$$\mathcal{P}(t) = \operatorname*{argmax}_p P(phase = p \mid \mathbb{O}_{1:t}) \quad (3)$$

$$= \operatorname*{argmax}_p \sum_{x_i} \phi_{x_i}(p) P(X_t = x_i \mid \mathbb{O}_{1:t}) \quad . \quad (4)$$

5. Observations

The multi-camera system provides sequences of 3D occupation grids. For phase recognition, we compute features which coarsely describe the spatial distribution of motion in the OR over time, without linking these to specific instances of persons and objects. We thereby rely on the fact that the different phases are discriminative with respect to different motion patterns appearing at key-locations in the OR. For instance, the *patient entering* (Ph. 1 in Fig. 1) can be identified by a strong motion close to the entrance of the OR, while the *surgery* phases are characterized through distinct motion patterns appearing in the neighborhood of the operation table. We experimented as well with occupancy based features for phase discrimination. Unfortunately, it revealed to provide poorly discriminative information, even in combination with motion features. Below, we only present the computation of the flow features.

5.1. 3D Motion Flow Features

For feature computation we split the reconstruction volume into a set of $Q \times R \times S$ evenly spaced cells **c**. For each cell, we compute a histogram of 3D motion orientations (cf Fig. 5). To avoid quantization effects at the boundaries of the cells and to introduce invariance to small variations, we use a smooth voting scheme for histogram computation based on radial basis functions. In detail, from the occupation grid sequences we compute 3D optical-flow sequences:

$$\{\mathbf{f}_{1:T}\}, \quad \mathbf{f}(\mathbf{v}) : \Omega \to \mathbb{R}^3, \quad \Omega \subset \mathbb{R}^3 \quad , \tag{5}$$

using the Lucas-Kanade method applied in 3D [2]. The flow vectors $\mathbf{f}(\mathbf{v})$ for all voxels $\mathbf{v}$ are then quantized into n orientation bins $\tilde{h}_1(\mathbf{v}), \ldots, \tilde{h}_n(\mathbf{v})$ using a soft voting scheme, and accumulated into histograms $\{\mathbf{H}_{1:Q \cdot R \cdot S, 1:n}\}$, one for each of the $Q \times R \times S$ evenly spaced cells $\mathbf{c}_i$:

$$\mathbf{H}_{ij} = \sum_{\mathbf{v} \in \Omega} \tilde{h}_j(\mathbf{v}) \exp\left(-(\mathbf{v} - \bar{\mathbf{c}}_i)^2 / 2\sigma_i^2\right) \quad , \tag{6}$$

where $\bar{\mathbf{c}}_i$ is the centroid of cell $\mathbf{c}_i$, and σ_i controls the radius at which flow vectors contribute to a certain cells.

To quantize the 3D flow vectors we adapt a recently proposed approach [8], which uses regular polyhedrons to define the histogram bins. This quantization scheme avoids well-known problems of standard spherical quantizations. Given a regular n-sided polyhedron with facet normals $\{\mathbf{p}_{1:n}\}$, a flow vector $\mathbf{f}(\mathbf{v})$ votes into each bin as follows:

$$h_i(\mathbf{v}) \quad = \quad max\left(\frac{\mathbf{f}(\mathbf{v})^{\mathrm{T}} \cdot \mathbf{p}_i}{\|\mathbf{f}(\mathbf{v})\|_2} - q, 0\right) \quad . \tag{7}$$

q is chosen such that a flow vector lined up with one of the bin's normals will only vote into this bin, *i.e.* $q = \mathbf{p}_j^\top \mathbf{p}_k$, with $\mathbf{p}_j, \mathbf{p}_k$ being direct neighbors. Otherwise, it votes into several neighboring bins, in proportion to its closeness to the corresponding bin.

The final contribution of flow vector $\mathbf{f}(\mathbf{v})$ into histogram bin $\tilde{h}(\mathbf{v})_i$ is computed by normalizing the values of $h_i(\mathbf{v})$ to the unit and weighting them by the flow vectors magnitude:

$$\tilde{h}(\mathbf{v})_i = \frac{\|\mathbf{f}(\mathbf{v})\|_2 \cdot h_i(\mathbf{v})}{\sum_i h_i(\mathbf{v})} \quad . \tag{8}$$

In practice, we use a dodecahedron with 12 bins for the quantization. The volume is split into a grid of $3 \times 3 \times 2$ cells and we set $\sigma_i = r_i * 0.6$, where r_i is the average spacing between cell $\mathbf{c}_i$ and its neighbors. The vectors of size $12 \times 3 \times 3 \times 2$ obtained by vectorization of the histograms are passed as observations $\mathbb{O}_{1:T}$ to the recognition system after dimensionality reduction using PCA.

6. Experimental Validation

To validate our method, we recorded in a mock-up OR different instances of the surgery workflow described in Fig. 1, performed by actors familiar with the OR. The scene

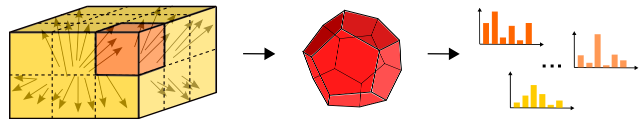

Figure 5. Computation of 3D flow histograms within the reconstructed volume, using regular polyhedron quantization.

contained simultaneously up to three persons, three tables and a ceiling OR light. The room where the activities take place is observed by a multi-camera system consisting of 9 synchronized and calibrated cameras fixed to the ceiling of the room. The acquisition, as well as the computation of the background subtraction and the 3D reconstruction are distributed over a small computer network (1 server, 3 clients), to achieve the required real-time execution. The resulting data-set is composed of a series of 22 videos (illustrated in Fig. 2) of 3D volumetric reconstructions of resolution 128x128x128. All videos are acquired with 15 frames/s.

We conducted several experiments to evaluate the phase recognition results on-line and off-line, as well as with and without final computation of the phase probabilities after global EM. The utility of the phase probabilities to track semantic shift appears in particular in experiments conducted with sub-sets of annotated sequences and sub-sets of the labeled phases to be recognized. We also compare WHMM with two methods that do not use the temporal constraints of the workflow. In the first method used for comparison, named MAP-HMMs, all phases are modeled independently by different HMMs trained on sub-windows of the data. Maximum likelihood classification is performed at each time-step using a sliding-window. In the second method, sub-HMMs for all phases are arranged in parallel and connected via a sub-HMM modeling background activity. We use this form of connections instead of fully inter-connected HMMs, having a single or multiple HMMs modeling the background activities, since it showed better results. We call this approach CO-HMMs. For evaluation, we perform a full cross-validation. The presented results are the mean over all tests performed with the leave-one-out method.

6.1. Error Measures

We use 5 types of error measures to present the recognition results on a sequence. All these errors apply for both on-line and off-line recognition. *Accuracy* indicates the percentage of correct detections compared to the ground truth in the complete sequence. To account for variations in length between the phases, we also define the following errors computed per phase: *Recall* (or correct positives) is the number of true positives divided by the total number of positives in the ground truth. *Correct negatives* is the number of true negatives divided by the total number of negatives in the ground truth. *Precision* is the number of true positives divided by the number of true and false positives. *Overall* *success* is the number of true positives and true negatives divided by the length of the sequence. Correct positives, correct negatives and overall success are inspired from the error measures used in [17].

6.2. Phase Recognition Results

Summarizing results are displayed in table 1. As expected, on-line results are slightly worse than off-line results since less information is available. WHMM outperforms both CO-HMMs and MAP-HMMs approaches, showing the importance of using all information provided by the annotation. A difficulty when performing on-line recognition with MAP-HMMs on phases with highly varying lengths, is the choice of the window size. We performed experiments with different sizes but could not match the results of the two other approaches. CO-HMMs perform worse than WHMM as they are less constrained and do not take advantage of all temporal relationships, even after EM training. An additional disadvantage of CO-HMMs are the numerous short and incorrect transitions that occur both off-line and on-line. This effect also occurs on-line for WHMMs, but in a much smaller scale, as shown in figure Fig. 6. This figure shows the number of transitions occurring between all pairs of phases during a complete cross-validation test. Self-transitions on the diagonal were removed for better visualization and the background phase is not taken into account. CO-HMMs perform 4 times more short incorrect transitions than WHMMs.

The influence of EM training and of phase probabilities is shown in Fig. 7. We see that performing the phase probability computation after EM training improves the results. Additionally, this figure shows how the overall results vary depending on the percentage of labeled sequences available in the training set. For this experiment, results were computed from a single random split of the sequences into training and test data for each number of labeled sequences and cross-validation test. Results are expected to get smoother if they are averaged on all the possible subsets.

Results using only a subset of the labels are shown in table 2 . In this experiment, only four labels are available in the training sequences, corresponding to phases 3,6,7 and 8. WHMMs clearly outperform in this case CO-HMMs. Phase probabilities also improve significantly the results, as a semantic shift during EM can easily occur in the long background phases.

On-line results for each phase are given for WHMMs in table 3. This table shows reliable detection using 3D-flow, except for phase 5. Interestingly, this corresponds to an emergency, which is actually an anomaly. Indeed, it consists of an accelerated performance of the first preparation phases and only occurs 4 times in our dataset. When wrongly detected, it is recognized as these similar phases.

		Accuracy		Recall		Precision	
		W	CO	W	CO	W	CO
OF	NO	90.5	71.4	88.4	61.6	87.4	60.0
	EM	86.6	66.0	84.8	56.5	73.9	51.9
	PV	92.9	71.0	91.3	57.2	75.8	56.4
ON	NO	78.5	71.1	70.5	62.2	66.1	56.1
	EM	80.7	65.7	75.3	56.8	71.5	49.8
	PV	84.2	70.7	78.8	57.8	74.8	55.1

Table 2. Results in percent using solely labels 3, 6, 7 and 8. Comparison between WHMMs (W) and CO-HMMs (CO), on-line (ON) and off-line (OFF), without EM (NO), with EM only (EM) and with EM followed by computation of phase variables (PV).

Phase	#	Recall	Preci	CorNeg	OveSuc
(B) Backgr.	22	91.1	85.8	94.2	93.5
(1) Pat. Enter.	18	86.0	76.0	99.8	99.5
(2) Anaesth.	5	86.2	96.8	99.8	98.8
(3) Pat. Prep.	18	86.5	89.6	99.3	97.5
(4) Surg. Prep.	18	89.7	94.6	99.5	98.6
(5) Emerg.	4	15.8	91.5	99.9	96.8
(6) Min. Inv.	11	85.7	76.9	99.6	96.9
(7) Open Surg.	15	89.3	68.8	97.0	95.4
(8) Suturing	22	87.3	86.1	98.4	97.1
(9) Pat. Leav.	22	94.8	87.2	99.7	99.6
(10) Cleaning	22	96.0	99.0	99.9	99.5

Table 3. On-line results in percent presented for all phases using WHMMs. Column *Phase* indicates the phase label, as in Fig. 1. B stands for the phase modeling background activity. Column # shows the number of occurrences of each phase within the dataset.

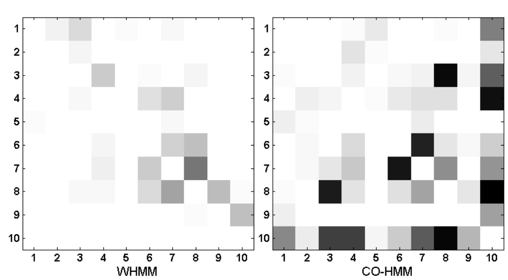

Figure 6. Occurring phase transitions, on-line, for WHMMs and CO-HMMs. White color means absence of transition.

7. Conclusion

In this paper, we addressed the problem of monitoring environments constrained by an overall and repetitive workflow containing alternatives, such as operating rooms and production lines. We proposed to capture the activity in such complex environments with 3D-motion flow, obtained in real-time using videos from a multi-camera system. To perform reliable on-line recognition of the phases, we introduced WHMMs, a form of HMMs augmented with phase-probabilities which models the complete workflow.

	Accuracy			Recall			Precision			CorNeg			OveSuc		
	W	CO	MA	W	CO	MA	W	CO	MA	W	CO	MA	W	CO	MA
OFF	92.5	79.4	70.5	92.2	80.7	63.6	89.9	72.8	53.5	99.3	97.4	96.3	98.5	95.6	96.6
ON	89.2	78.2	70.5	87.8	79.5	63.6	85.3	62.3	53.5	98.8	97.3	96.3	97.5	95.4	96.6

Table 1. Summarized results in percent, comparing WHMMs (W), CO-HMMs (CO) and MAP-HMMs (MA), on-line (ON) and off-line (OFF), using EM and computation of phase probabilities.

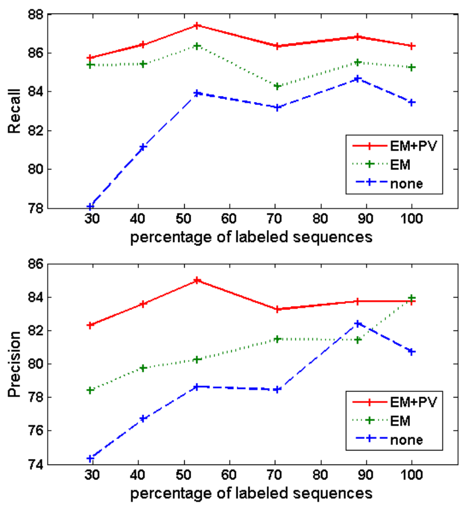

Figure 7. Precision and Recall of WHMMs as a function of the percentage of annotated sequences in the training set. On-line results before global EM (none), after global EM (EM) and after global EM and computation of phase variables (EM+PV).

Off-line and on-line results, as well as comparison to other approaches have shown the advantages of WHMM over simpler methods that do not fully use the information provided by the labeled sequences. The proposed approach was demonstrated on sequences from a medical application. Given the generality of the features, it could also be applied to other workflow processes.

In future work, we plan to learn a more efficient model, by allowing the phases to share parts of their structure. This can prove especially useful in cases where the labeled sequences contain partial and complementary information. We will also focus on recognizing additionnal semantic information, like the number of persons and their role, using high-level domain descriptions such as ontologies

Acknowledgments: This research was partially funded by Siemens Healthcare, the german excellence cluster CoTeSys and the european project PHAROS. The authors thank A. Ladikos, A. Ahmadi and J. Deville for helping with the data acquisitions.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU'99*.

[2] K. M. C. Simon Baker, Raju Patil and I. Matthews, *Lucas-kanade 20 years on: Part 5*, Tech. Report CMU-RI-TR-04-64, 2004.

[3] A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Proceedings of the Royal Society*, 352:1257–1265, 1997.

[4] K. Cleary, H. Y. Chung, and S. K. Mun. Or 2020: The operating room of the future. *Laparoendoscopic and Advanced Surgical Techniques*, 15(5):495–500, 2005.

[5] A. A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV'03*.

[6] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *CVPR'98*.

[7] R. Hamid, S. Maddi, A. F. Bobick, and I. A. Essa. Structure from statistics: Unsupervised activity analysis using suffix trees. In *ICCV'07*.

[8] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC'08*.

[9] K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darrell. Activity zones for context-aware computing. In *In UbiComp'03*.

[10] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR'07*.

[11] D. Moore and I. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *AAAI'02*.

[12] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.

[13] N. Nguyen, D. Phung, S. Venkatesh, and H. H. Bui. Learning and detecting activities from movement trajectories using the hierarchical HMMs. In *CVPR'05*.

[14] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *CVIU'04*.

[15] C. Pinhanez and A. Bobick. Intelligent studios modeling space and action to control tv cameras. *AAI'97*.

[16] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE'89*.

[17] Y. Shi, A. Bobick, and I. Essa. Learning temporal sequence model from partially labeled data. In *CVPR'06*.

[18] V.-T. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *IJCIA'03*.

[19] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV'07*.

[20] T. Xiang and S. Gong. Optimising dynamic graphical models for video content analysis. *CVIU'08*.