POSTER: Planning-based Workflow Modeling for AR-enabled Automated Task Guidance

Hao Zhang*, William Hoff*
Department of Computer Science, Colorado School of Mines, Golden, CO 80401

ABSTRACT

In this paper, we implemented and validated a workflow modeling approach that is able to model a sequence of procedures to achieve a complex task to enable an AR-based automated task guidance system. We formulated automated task guidance as a decision making problem, based upon the general Partially Observable Markov Decision Processes (POMDP) paradigm as the foundation. Our approach is able to provide actionable information to actively instruct users through a complex multi-step task. Our method can also plan ahead an action sequence that is optimal in the long term, while maintaining flexibility to deal with changes in an uncertain environment. We validated our approach in the applications of copy machine inspection and compressor startup guidance. The experimental results have shown the effectiveness of our planning-based workflow models in real-world applications.

Keywords: Workflow modeling, augmented reality, task guidance.

Index Terms: Human-centered computing → Mixed / augmented reality

1 INTRODUCTION

Maintenance is an typical task in industrial and daily life. To deal with the problem of maintenance errors and lack of expertise, what is needed is an automated maintenance guide, to actively guide novice users through the task, and monitor the user's actions to check if all steps are being followed. Recently, augmented reality (AR) has shown great potential to improve the effectiveness of personnel in performing maintenance tasks, resulting in improved results in terms of reduced task time and reduced number of errors [2, 7, 8].

To enable an AR-based system for automated guidance, modeling the workflow of a maintenance task is critical. The workflow can be defined as a sequence of actions necessary to complete a well-defined complex task. Traditional approaches to model task workflow are typically based upon Hidden Markov Models (HMM) or Dynamic Bayesian Networks (DBN) [1, 5, 6], which however only encode the temporal relationship of object states (i.e., the configuration of the object determined by the presence or position of subparts). Thus, such approaches are typically not able to provide directly actionable information for task guidance.

In this paper, we develop and implement an alternative approach for modeling workflow to provide actionable information to enable AR task guidance for a wide variety of tasks. Different from HMM or DBN-based methods, our workflow modeling method is designed following a general Bayesian *planning* paradigm named Partially Observable Markov Decision Processes (POMDP) [4], which models time dependencies of both states and actions. We construct our *Planning-based Workflow Model* (PWM) using sequences of actions provided by domain experts and automatically recognized object states [3]. The contribution of this poster paper is the *implementation* of PWM and the *validation* of its effectiveness in two real-world automated task guidance applications.

*e-mail: {hzhang, whoff}@mines.edu

2 BACKGROUND FOR PWM

We formulate automated task guidance as a decision making problem. Specifically, following the general POMDP paradigm, the PWM is represented by a tuple $(S, A, T, R, \Omega, O, \gamma)$.

- S: a finite set of states that encode the configurations of a targe object needed to be maintained;
- A: a finite set of actions that can change the object configurations;
- T: transition function, a set of conditional transition probabilities between states. $T(s'|a,s)$ is the probability that the system will end up in state s after taking action a in state s;
- Ω: a finite set of observations from noisy sensors;
- O: observation function $O(o|s',a)$, which is a conditional observation probability that the system receives observation o after taking action a and getting into state s';
- $R : S \times A \to R$, such that $R(s,a)$ is the immediate reward received when the system takes action a in state s;
- γ: discount factor that considers long-term rewards.

Because the system does not directly know the true state of the object, it must make decisions under uncertainty of true states that need to be estimated from noisy sensory data. Through observing the object, PWM needs to update its belief in the true state through updating the probability distribution of the current true state. We define this probability distribution as *belief*, denoted as $b(s)$, which denotes the probability that the object is in state s. Then, PWM needs to update its belief upon taking action a and observing o. Since the state sequence is Markovian, maintaining a belief over them solely requires knowledge of the previous belief state, the taken action, and the current observation, which can be denoted as $b' = \tau(b,a,o)$. Below we describe how this belief update is computed.

After getting into s', PWM observes $o \in \Omega$ with the probability $O(o|s',a)$. Let b be a probability distribution over the state space S. $b(s)$ denotes the probability that the object is in state s. Given $b(s)$, after taking action a and observing o, we obtain:

$$b'(s') = \eta O(o|s',a) \sum_{s \in S} T(s'|s,a) b(s), \tag{1}$$

where $\eta = \left(\sum_{s' \in S} O(o|s',a) \sum_{s \in S} T(s'|s,a) b(s) \right)^{-1}$ is the normalizing constant.

A policy π specifies an action $a = \pi(b)$ for any belief b. Our objective is to choose a policy that is able to maximize the expected reward. The reward function over the belief state distribution can be computed by:

$$r(b,a) = \sum_{s \in S} b(s) R(s,a) \tag{2}$$

Then the expected reward $V^{\pi}(b_0)$ for policy π starting from belief b_0 is defined as:

$$V^{\pi}(b_0) = \sum_{t=0}^{\infty} \gamma^t r(b(t), a(t)) \tag{3}$$

The optimal policy $\pi^\star$, which provides actionable information, can be obtained by optimizing the long-term reward:

$$\pi^\star = \arg\max_{\pi} V^\pi(b_0) \tag{4}$$

PWM is able to estimate real object states from noisy sensory data and provide actionable information to actively instruct users to physically perform a maintenance task. PWM can also plan ahead with action sequences that are optimal in the long term, while maintaining flexibility to deal with changes in an uncertain environment.

As the major contribution of this paper, in the next two sections, we will present and discuss the implementation and validation of PWM in real-world daily-life and industrial applications.

3 Application 1: Copy Machine Maintenance

In this daily-life application, our objective is to create a PWM for an AR-based automated system that can guide users to inspect and find out which part of a copy machine has a jam. The copy machine we used in our experiment is illustrated in Figure 1. There are two places that may cause jam: top and side places. (1) Top: There are two doors in the top place and the object state depends on the status of the doors. A typical inspection procedure suggested by the user manual for the top place is: all doors closed $\rightarrow$ open door 1 $\rightarrow$ check jam at door 1 $\rightarrow$ open door 2 $\rightarrow$ check jam at door 2. (2) Side: There are four doors in the side place. A typical inspection procedure for the side is: all doors closed $\rightarrow$ door 1 open $\rightarrow$ door 2 open $\rightarrow$ check jam at door 2 $\rightarrow$ door 3 open $\rightarrow$ door 4 open $\rightarrow$ check jam at door 4.

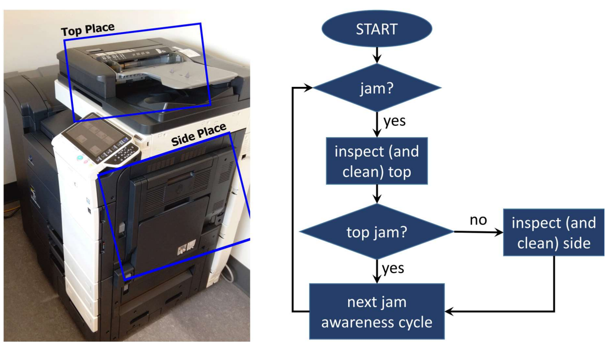

Figure 1: LEFT: One of the four copy machines used in the experiment. Two places may have a jam: top and side, which are marked by blue rectangles, respectively. RIGHT: Maintenance procedure of the copy machine, provided by the product manual.

When a jam happens, a user needs to inspect both places: top and side places. We assume the jam can only appear in one place, which is reasonable since one paper cannot be jammed in two separate places. Once the jam is identified and cleared in one place, the inspection procedure ends, which means we do not have to examine the remaining places any longer. The overall maintenance procedures of the copy machine are shown in Figure 1 (RIGHT). Each place inspection is formulated as an individual PWM.

3.1 PWM-based System Modeling

In this jam inspection application, the PWM can be used to model the procedures of inspecting the top or side place. Since inspecting both places has a similar procedure, here we only use the top place as an example to describe our implementation details. Specifically, the implementation is as follows:

Action Space A: $A = \{\text{open door } i, \text{close door } i, \text{inspect (and clean)}\}$, where $i = \{1, 2\}$, includes all valid actions in top jam inspection.

State Space S: In the top jam inspection task, the copier has three physical statuses: *all doors closed, door 1 open, door 1 & 2 open*.

Algorithm 1: State Transition Learning

Input : Training data, state space S
Output: State transition T

```
Initialize: State transition map STM, whose keys are states in
  the state space S, demonstration sequence index
  i_d = 1, i_d ∈ {1,2,··· ,n_d}, state index in each demonstration
  sequence i_ds = 1, i_ds ∈ {1,2,··· ,n_ds};
/* Establish the state transition map STM */
while i_d ≤ n_d do
    i_ds = 1;
    while 1 < i_ds ≤ n_ds do
        Append the value of key s(i_ds − 1) with
          (a(i_ds − 1), s(i_ds));
        i_ds++;
    end
    i_d++;
end
/* Calculate the state transition T */
for key s in STM do
    T(s,a,s') = # of (a,s') in STM[s] / # of (a) in STM[s]
end
return state transition T.
```

So we need two bits to encode these statuses: $[0,0]^\top$ represents the status *all doors closed*, $[1,0]^\top$ denotes the status *door 1 open*, $[1,1]^\top$ denotes the status *door 2 open*. Since the copier's door 2 is designed to be open only after door 1 is already open, the status $[0,1]^\top$ is not possible.

Since a jam may happen in either of the two doors, we need to inspect each of them. In order to encode whether each door has been inspected, we introduce two additional bits to represent the inspection status: '0' means the jam has not been inspected, and '1' denotes it has been inspected.

After removing impossible statuses (constrained by the copier's hardware), the final state representation in PWM for top jam inspection is listed in Table 1, which contains seven states $S = \{s_0, \ldots, s_6\}$, corresponding to different door and inspection statuses.

State Transition T: Given the demonstrations provided by domain experts, which are processed using our vision-based state recognition system [3], they are converted to training data with a format described in Figure 2. Then, the state transition is learned by Algorithm 1 using the training data. An illustration of the learned state transitions and their probabilities is shown in Figure 3.

Frame	Figure name	Status (ground truth)	Prob(o1)	Prob(o2)	Prob(o3)
0	Img_1_0.png	1	0.3671291	0.2967421	0.3361280
1	Img_1_1.png	1	0.3637380	0.2974490	0.3388130
2	Img_1_2.png	1	0.3952390	0.2976010	0.3071600

Figure 2: Format of training data from our previous vision-based state recognition system [3].

Observation Space Ω: The observation space $\Omega = \{$*all doors closed, door 1 open, all doors open*$\}$, which are the physically possible door statuses that can be distinguished by our previous vision-based state recognition system [3]. Examples of each door status are provided in Figure 4.

Table 1: The state space used in the subtask of top jam inspection

Notation: In "Inspection Status", '00' represents neither door 1 nor door 2 are inspected, '10' represents only door 1 is inspected, and '11' represents both door 1 and door 2 are inspected.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6
State Representation	$\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}1\\0\\1\\0\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\0\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1\\0\\1\\1\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\1\end{bmatrix}$
Door Status Inspection Status	all doors closed not inspected	door 1 open not inspected	door 1 open door 1 inspected	door 2 open door 1 inspected	door 2 open both inspected	door 1 open both inspected	all doors closed both inspected

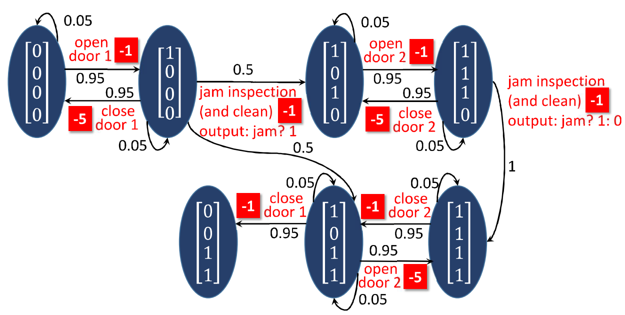

Figure 3: State transition for top jam inspection. Each node in the model represents a state in PWM. In state $[a,b,c,d]^\top$, $[a,b]^\top$ indicates the status of two doors in the copy machine's top place, '1 represents *open* while '0 denotes *close*; $[c,d]^\top$ indicates the inspection status. '00 means neither of the two doors are inspected, '10' represents only door 1 is inspected, and '11' denotes both two doors are inspected. Red strings are actions in each state. Black numbers are the state transition probability after taking an action in each state. White numbers within red squares are rewards with respect to each state-action pair. This figure is best viewed in color.

Observation Matrix O: The observation matrix indicates the probability that the vision system observes o_j when the copier is in state s_i, that is, $O(i,j) = \Pr\{o_j|s_i\}$. $O(i,j)$ is calculated by the equation $O(i,j) = \frac{1}{n}\sum_{k=1}^{n}\Pr\{o_j|s_i\}$, where n is the number of instances that the vision system observes o_j when the copier machine is in state s_i.

Reward R: The reward assignment mechanism in both top and side place inspection are the same. The reward for opening doors before inspection and closing doors after inspection is -1, which matches our intention. However, we penalize inappropriate actions at each state with the reward -5, that is, the reward for opening doors after inspection as well as closing doors before inspection is -5. The reward for inspection is -1. The detailed reward assignment in the top inspection scenario can be found within the red boxes in Figure 3. It should be pointed out that the reward of state-action pairs that are not in the transition diagram is -10, forcing the large penalty for invalid actions. The reward discount constant γ is set to 0.5.

3.2 Experimental Results

During the learning phase using training data, T is learned to be $T(s,a,s') = 5.6\%, T(s,a,s) = 94.4\%, s,s' \in S$ and $s \neq s'$, and the observation matrix O of the top inspection is learned to be:

$$\begin{bmatrix} 0.4100 & 0.2800 & 0.2900 \\ 0.2900 & 0.4000 & 0.3200 \\ 0.3000 & 0.3200 & 0.3900 \end{bmatrix} \tag{5}$$

After the learning process to build the PWM, we are able to deploy it to guide users on the jam inspection task. A real-world test result

Figure 4: Exemplary training images of different door statuses for the copier top jamming case. Each row represents one door statuses, and from top to bottom they are: (a) all doors are closed; (b) door 1 is open; (c) all doors are open.

can be found in Appendix I, where the output text are the actionable instructions provided to the users, helping them to inspect jams of the copier correctly and smoothly.

The quantitative performance comparison of our PWM with baseline methods for jam inspection guide is illustrated in Figure 5. It is observed that the method based on direct observations from the vision system (i.e., without modeling the states) is not stable and accurate enough, and there are a few wrong identification of the statuses. However, PWM is able to filter out noise and obtain much stable belief estimation. If the underlying POMDP paradigm is not utilized in a workload model, the system typically cannot provide actionable instructions. Also, if a Markovian assumption is not assumed, the instructions provided to users may change frequently, which could confuse the users and decrease the user satisfaction to automated maintenance guide systems.

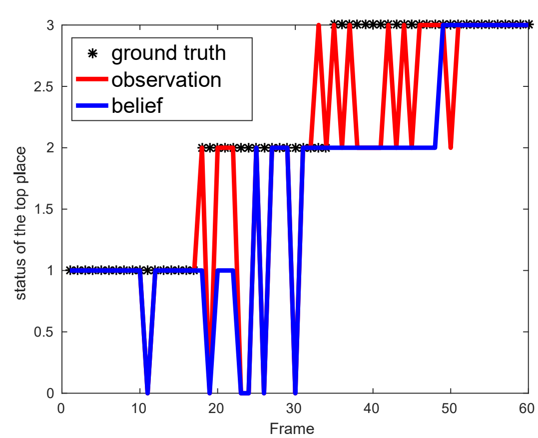

Figure 5: Performance comparison of PWM with baseline methods for top jam inspection. There are three physical status of the top place: $status_1$ = all doors closed, $status_2$ = door 1 open, $status_3$ = door 2 open. $status_0$ means the current status is invalid.

4 Application 2: Compressor Inspection

The second real-world application is inspired by the industrial need to inspect a compressor in the underground. In this experiment, the used compressor is located in the Edgar mine owned and managed by Colorado School of Mines. Figure 6 illustrates the appearance of the compressor. There exist multiple steps throughout the compressor inspection. Several steps are sequential, meaning these steps must be finished in a strict order, while other steps are non-sequential and their order is not important.

4.1 User-Centric Model Construction

In the jam inspection guide problem, we (the model designer) manually defined our workflow models, including states, actions, observations, and rewards. However, due to the limited domain knowledge of model designers who are typically computer scientists, we typically cannot create a comprehensive model that captures all domain considerations. Or it could be too time-consuming and unaffordable to discuss all detailed issues with domain experts.

In order to make our PWM applicable and scalable to a variety of tasks in different domains, we propose to directly learn our model autonomously using the expert demonstrations only. Our user-centric model construction approach does not depend on specific knowledge of model designers, and the learned PWM is determined completely by the input training data. In this case, we still follow the POMDP paradigm described in Section 2 as the foundation to build our user-centric PWM, due to its capability in sequential decision making under uncertainty. The difference is that the parameters of PWM and POMDP are now completely determined by expert demonstrations only instead of depending on the model designer's knowledge, i.e., we (model designers) do not assume to have any prior knowledge of the target system to be inspected or repaired.

In our user-centric model building approach, the ***state space*** is determined by the domain experts who have prior knowledge of the target system to be inspected or repaired. The experts provide the state information through a log file. Then, our model reads in this file and establishes the states autonomously. The user-centric PWM determines the ***action space*** in the same fashion, without requiring the model designer to have any task-specific knowledge.

The ***observation space*** is built through learning from a training data file, which is generated by the vision system [3] according to the expert demonstration videos, with a similar format to the one in Figure 2. After obtaining the observation space, we need to learn the observation matrix O, which represents the probability of observing o_i when the object is in state s_j. The learning process is similar to that in the previous jam inspection task. The main difference is that the observation space also needs to be learned in this case, which is calculated by:

$$O(i,j) = \frac{1}{N}\sum_{k=1}^{n} \Pr\{o_j|s_i\}, \tag{6}$$

Figure 6: The compressor in the Edgar underground mine is used in this experiment.

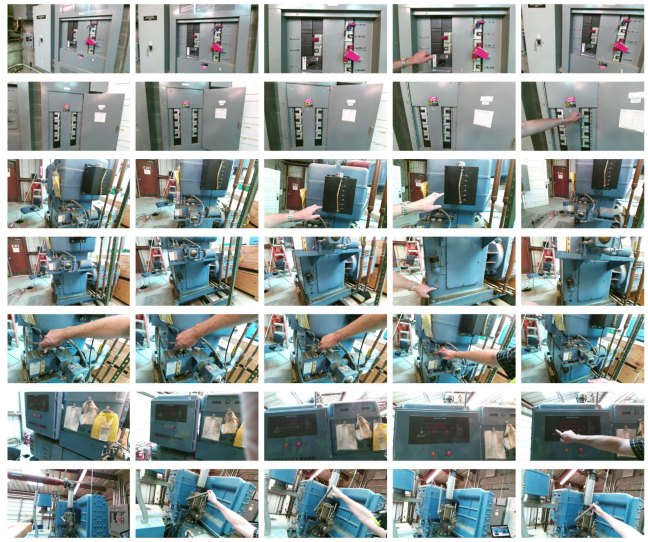

Figure 7: Examples of observations in different states for compressor startup guidance. Each row represents one state; from top to bottom they are: (a) panel 1; (b) panel 2; (c) upper oil reservoir; (d) lower oil reservoir; (e) frame oil buttons; (f) front panel; (g) back lever.

where N is the number of instances that the vision system observes o_j when the target object is in state s_i. The examples of observations in seven states in the compressor inspection application are shown in Figure 7. Using the same training data, the ***state transition*** can also be learned using Algorithm 1.

4.2 Experimental Results

In the experiments, we assume that *multiple* domain experts provide demonstrations of finishing the compressor inspection task as the training data. Since each of the experts may have their own inspection preference, the sequence of actions provided by the experts could be different, as shown in Table 2 in which four experts provide three different sequences of completing the task. The sequential variations in the demonstrations can greatly increase the learning difficulty, which however is common in real-world applications.

Our user-centric PWM can well address this issue, which learns the following state transition:

$$\begin{bmatrix}
0.875 & 0.063 & 0.031 & 0.031 & 0 & 0 & 0 \\
0 & 0.909 & 0.068 & 0.023 & 0 & 0 & 0 \\
0 & 0.018 & 0.929 & 0.036 & 0.018 & 0 & 0 \\
0 & 0.019 & 0 & 0.923 & 0.058 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.917 & 0.083 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.958 & 0.042 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.000
\end{bmatrix}$$

The observation matrix in the compressor inspection application is learned as follows:

$$\begin{bmatrix}
0.29 & 0.11 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 \\
0.10 & 0.28 & 0.12 & 0.10 & 0.10 & 0.10 & 0.10 & 0.10 \\
0.10 & 0.10 & 0.28 & 0.13 & 0.10 & 0.10 & 0.10 & 0.10 \\
0.10 & 0.10 & 0.10 & 0.28 & 0.12 & 0.10 & 0.10 & 0.10 \\
0.10 & 0.10 & 0.10 & 0.10 & 0.28 & 0.12 & 0.10 & 0.10 \\
0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.28 & 0.12 & 0.10 \\
0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.12 & 0.28 & 0.10
\end{bmatrix}$$

The number of observations is also seven (as they are defined as the estimation of the seven states in our experiment). The dimension of the observation matrix O is 8×8 is because we intentionally add an additional observation called "unseen observation", which is used to indicate the intermediate status during state transition, which are undefined and considered as an "unseen observation".

Table 2: Different inspection sequences provided by four experts.

Expert	Inspection sequence, each number denotes a state
1	$1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$
2	$1 \to 3 \to 2 \to 4 \to 5 \to 6 \to 7$
3	$1 \to 4 \to 2 \to 3 \to 5 \to 6 \to 7$
4	$1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$

After learning all the parameters in our user-centric PWM, the model can be used to guide users to perform compressor inspection. A test case is demonstrated in Appendix II, where the model outputs provide actionable instructions to the users. Comparison of quantitative performance of our model with baseline methods is also illustrated in Figure 8.

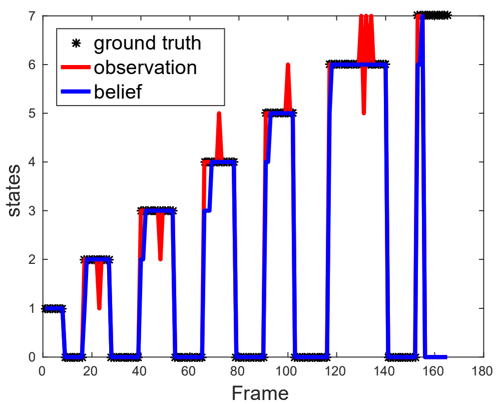

Figure 8: Performance comparison in the application of compressor inspection. There exist seven valid states, and *state* 0 indicates that the current state is invalid.

In this user-centric setup with multiple domain experts providing demonstrations, the conflicts of demonstrations from different users in the training data can be well addressed by our user-centric PWM through increasing the connectivity among the states. In addition, this data-driven, learning-based model construction approach allows domain experts to use our model without knowing technical details about our approach. As long as the experts provide the prior knowledge of a specific task, and our user-centric PWM can transform the knowledge into the model autonomously.

5 Conclusion and Future Work

This paper describes the implementation and validation results of a planning-based workflow modeling approach based on the Markov decision process. Different from approaches based on HMMs and DBNs, the PWM is able to provide actionable information as direct instructions of next actions to take. We validate our approach in the daily-life copier inspection application, as well as in the industrial application to inspect underground compressors where we develop a user-centric model that separates model design and domain expertise. Both validation scenarios have shown the effectiveness of planning-based workflow modeling.

In our current prototype, the state definition is completely determined by humans (either model designers or domain experts), which could contain significant ambiguity. As a future work, we will develop data-drive state learning methods to automatically identify and adapt states. In addition, we will integrate this PWM with our previous vision system [3] onto a physical hardware (e.g., a DAQRI helmet) and validate its overall effectiveness in realtime scenarios.

Acknowledgments

This work was partially supported by a gift from DAQRI, LLC.

References

[1] T. Blum, H. Feußner, and N. Navab. Modeling and segmentation of surgical workflow from laparoscopic video. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2010.

[2] S. Henderson and S. Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 17(10):1355–1368, 2011.

[3] W. Hoff and H. Zhang. Learning object and state models for ar task guidance. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2016.

[4] M. J. Kochenderfer and H. J. D. Reynolds. *Decision making under uncertainty: theory and application*. MIT press, 2015.

[5] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[6] R. Liu, X. Zhang, and H. Zhang. Web-video-mining-supported workflow modeling for laparoscopic surgeries. *Artificial intelligence in medicine*, 74:9–20, 2016.

[7] N. Petersen, A. Pagani, and D. Stricker. Real-time modeling and tracking manual workflows from first-person vision. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[8] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche. An augmented reality training platform for assembly and maintenance skills. *Robotics and Autonomous Systems*, 61(4):398–403, 2013.

Appendix I: Automated Maintenance Guide for Copier's Top Jam Inspection

Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Invalid perception result, re-perceiving now
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Please open door 1
Invalid perception result, re-perceiving now
Please open door 1
Please open door 1
Please open door 1
Invalid perception result, re-perceiving now
Invalid perception result, re-perceiving now
Please inspect the jam at door 1 here
Is there jam here? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 0`
Invalid perception result, re-perceiving now
Please open door 2
Please open door 2
Please open door 2
Invalid perception result, re-perceiving now
Please open door 2
Please open door 2

Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please open door 2
Please inspect the jam at door 2 here
Is there jam here? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 2
Please close door 1
Please close door 1
Please close door 1
Please close door 1
Please close door 1
Please close door 1
Please close door 1
Please close door 1

APPENDIX II: AUTOMATED MAINTENANCE GUIDE FOR COMPRESSOR INSPECTION

Please turn on the switch if it is off
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to PP2
Please move to PP2
Please move to PP2
Please move to PP2
Please move to PP2
Please move to PP2
Please move to PP2
Please move to PP2
Please turn on the switch if it is off
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please move to The upper oil reservoir
Please check oil level
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please move to The lower oil reservoir
Please check oil level
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Please move to The frame oil system
Plese press two buttons
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please move to The front of compressor
Please hit green Start button, then wait 2 minutes, and hit constant speed control button afterwards
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please move to The back of compressor
Please switch the lever position, and log in the start/stop book
Action finished ? Answer '1' for 'yes', '0' for 'no':
`(user interaction): 1`
Compressor Check Finished!