

# Automated Handwashing Assistance For Persons With Dementia Using Video And A Partially Observable Markov Decision Process

Jesse Hoey<sup>a,\*</sup> Pascal Poupart<sup>c</sup> Axel von Bertoldi<sup>b</sup>  
Tammy Craig<sup>b</sup> Craig Boutilier<sup>d</sup> Alex Mihailidis<sup>b</sup>

<sup>a</sup>*School of Computing, University of Dundee*

<sup>b</sup>*Dept. of Occupational Science and Occupational Therapy, University of Toronto*

<sup>c</sup>*David R. Cheriton School of Computer Science, University of Waterloo*

<sup>d</sup>*Dept. of Computer Science, University of Toronto*

---

## Abstract

This paper presents a real-time vision-based system to assist a person with dementia wash their hands. The system uses only video inputs, and assistance is given as either verbal or visual prompts, or through the enlistment of a human caregiver's help. The system combines a Bayesian sequential estimation framework for tracking hands and towel, with a decision theoretic framework for computing policies of action. The decision making system is a partially observable Markov decision process, or POMDP. Decision policies dictating system actions are computed in the POMDP using a point-based approximate solution technique. The tracking and decision making systems are coupled using a heuristic method for temporally segmenting the input video stream based on the continuity of the belief state. A key element of the system is the ability to estimate and adapt to user psychological states, such as awareness and responsiveness. We evaluate the system in three ways. First, we evaluate the hand-tracking system by comparing its outputs to manual annotations and to a simple hand-detection method. Second, we test the POMDP solution methods in simulation, and show that our policies have higher expected return than five other heuristic methods. Third, we report results from a ten-week trial with seven persons moderate-to-severe dementia in a long-term care facility in Toronto, Canada. The subjects washed their hands once a day, with assistance given by our automated system, or by a human caregiver, in alternating two-week periods. We give two detailed case study analyses of the system working during trials, and then show agreement between the system and independent human raters of the same trials.

*Key words:* Assistive technology, decision theory, human tracking, particle filters, behavior monitoring, POMDP, user trials.

## 1 Introduction

Older adults living with cognitive disabilities (such as Alzheimer’s disease or other forms of dementia) have difficulty completing activities of daily living (ADLs). Such older adults are usually assisted by a human caregivers, either family members or professionals, who monitor the activity in question and guide or prompt them when necessary. The dependence on a caregiver is difficult for the older adult, and can lead to feelings of anger and helplessness, particularly for private ADLs such as using the washroom. The demands and stress placed on caregivers is also significant due to the need for constant monitoring.

Computerized *cognitive assistive technologies (CATs)* are devices or systems that have the potential to allow this elderly population to complete such ADLs more independently by monitoring the users during the task, providing guidance or assistance when necessary, and in some cases alerting a caregiver when circumstances warrant. Ideally, such a system or device should be relatively non-invasive (e.g., not require special preparation for the user or constrain the user in any way), promote independent ADL completion (or partial completion) by the user, and relieve the caregiver of the burden of continual monitoring and guidance.

This paper presents a real-time system for assisting persons with dementia during handwashing, an important ADL that often poses difficulty with this population. The system described here was built upon three previous versions, each relaxing restrictive assumptions present in its predecessor [1,2]. In particular, the current system removes the need for worn markers to aid visual tracking (as used in [1]) and includes an explicit model of uncertainty not present in [2]. Our system for assisting persons with dementia during handwashing consists of four basic components, as shown in Figure 1. Video from a camera mounted above a sink is input to a system to track objects of interest (e.g., hands and towel). Object positions are passed to a *belief monitor* that estimates the progress of the user in the task as well as the user’s mental state (see below). This is represented as a *belief state* or distribution over possible task points and mental states. A *policy* then maps the belief state to a system action: this can be a no-op, an audio or video prompt, or a call for human assistance. The system action is fed back to the belief monitor to enable sequential updates of the belief state. Finally, the *prompter* plays the prompt or makes the assistance call as appropriate.

---

\* Corresponding author, Address: School of Computing, University of Dundee, Dundee, Scotland, DD1 4HN

*Email address:* [jessehoey@computing.dundee.ac.uk](mailto:jessehoey@computing.dundee.ac.uk) (Jesse Hoey).

*URL:* [www.computing.dundee.ac.uk/staff/jessehoey](http://www.computing.dundee.ac.uk/staff/jessehoey) (Jesse Hoey).

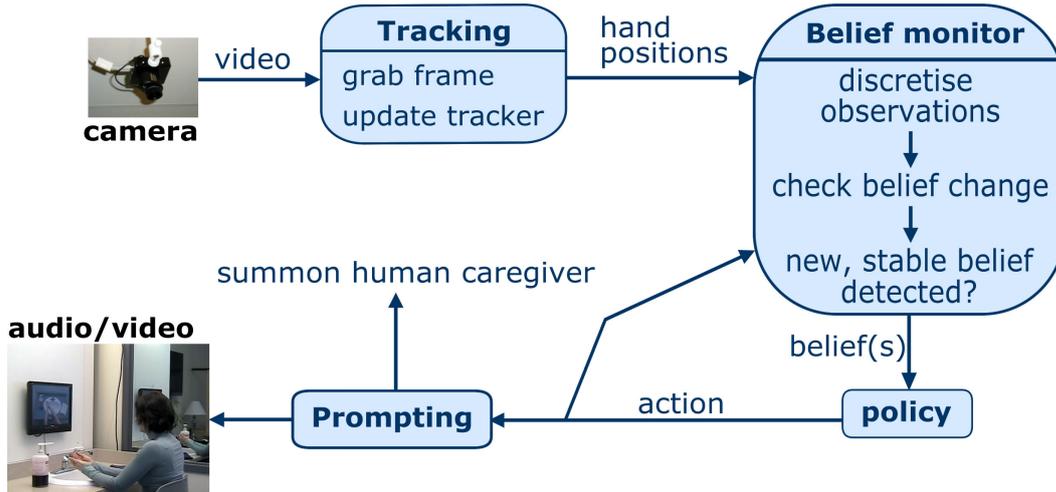


Fig. 1. Schematic of the system showing hand tracker, belief monitoring, policy, and prompting. The images show the laboratory test washroom with camera and video terminal.

A key feature of our system is the use of computer vision to ensure that monitoring is as *non-invasive* or as “passive” as possible. The user is not required to perform any special steps while performing an ADL, wear or activate a special device, or otherwise change their routines. Our approach has been motivated in part by the desire to see how far computer vision can be pushed to create such non-invasive assistive technologies.

For tracking, we use a mixed-state data-driven Bayesian sequential estimation method using *flocks* of color features [3], which allow objects to be robustly tracked over long periods of time, through large changes in shape and through partial occlusions. Our belief monitoring and policy systems use a *partially observable Markov decision process (POMDP)*. We show how the handwashing task can be modelled using a POMDP, with rewards and costs associated with specific actions and states reflecting their relative value (in terms of task progress, annoyance, inconvenience, etc.). The solution to the POMDP determines a decision policy—that associates actions with specific belief states—that optimizes long-term expected accumulated reward. The POMDP includes a model of the user’s mental state, such as responsiveness or awareness, and allows monitoring of these user traits. We denote these mental states in this paper as the user’s *attitude*. We use a heuristic for the temporal abstraction between tracking and behaviours.

Our previous work has demonstrated the effectiveness of the handtracker [3], a fully observable version of the POMDP model [2], and our initial tests with the current version of the system [4]. This paper expands on the work presented in [4] by showing results of the system performance taken from an eight-week trial of the system working with our target users: six persons with moderate to severe dementia in a long-term care facility in Toronto, Canada. The subjects

washed their hands once a day, with assistance given by our automated system, or by a human caregiver, in alternating two-week periods. We give two case study analyses of the system working during trials. We then show the rewards gathered by the system, based on human annotations of the true state during each trial, and we show how the system is able to estimate the user’s context and psychological state by comparing the system’s state with that given by independent human raters of the same trials. Our clinical findings (i.e. the effect of the system on users) are reported in [5].

The remainder of this paper is structured as follows. In Section 2, we give details about our computer vision system for tracking hands and towel. Section 3 discusses the POMDP, including specification, belief monitoring, temporal abstraction, and solution methods. Section 4 describes our experiments, and Section 5 shows our results both in simulations and in trials with our target users. Finally, we discuss related and future work and conclude.

## 2 Hand and towel tracking

Tracking an object in the presence of occlusions and distractions is a pervasive problem for computer vision applications. Objects to be tracked usually have some consistent features, are spatially compact, and move cohesively. Typical tracking methods use some model of the appearance of an object to be tracked, and estimate the fit of the model to the object over time. However, in many applications, the object’s shape and appearance may change over the course of a sequence. For example, human hands need to be tracked for many human-computer interaction tasks, but change shape and velocity fairly quickly, differences which must be accounted for. The method we present uses a generic type of model: a *flock* of features [6]. The features are characteristics of the local appearance of the object to be tracked, and they are loosely grouped using flocking constraints.

A flock consists of a group of distinct elements that are similar in appearance and that move congruously, but that can exhibit small individual differences. A flock has the properties that no member is too close to another member, and that no member is too far from the center of the flock. The flocking concept helps to enforce spatial coherence of features across an object, while having enough flexibility to adapt quickly to large shape changes and occlusions. The flock is only a very weak shape model, and adapts the distribution of its members to the current object distribution, which may even be non-contiguous (e.g., in the case of partial occlusions). To allow for very long-term tracking of multiple objects, we use three mixed-state particle filters [7], with data-driven proposals [8] to enable re-initialisation after a track is lost. We use one particle filter for each hand, and one for the towel. This method is robust to partial

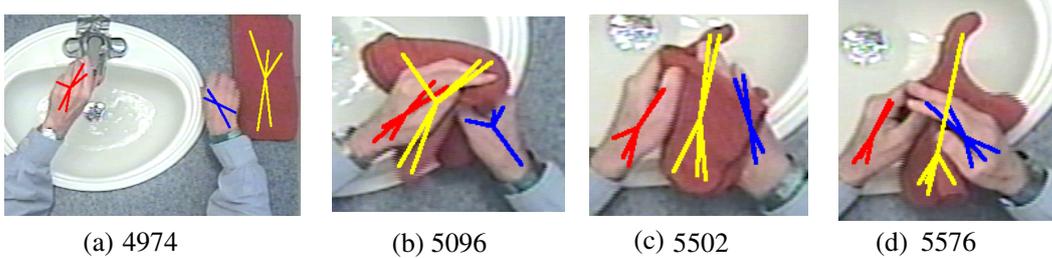


Fig. 2. Three flocks of 5 color features, or *specks*, tracking hands and towel. Images are labeled by frame number.

occlusions, distractors, and shape changes, and is able to consistently track objects over long sequences. Figure 2(a) shows three flocks of 5 color features tracking two hands and a towel. At the start, the members of each flock are distributed across the objects they are tracking. Figures 2(b)–(d) show the same three flocks later in the sequence, during occlusions and shape changes. The flocks maintain the track, even though the object shapes have changed.

## 2.1 Flocks of Features

More formally, a flock,  $\phi$ , is a tuple  $\{N_f, \mathbf{v}, \omega_f, \mathbf{W}, \xi_c, \xi_u\}$  where  $N_f$  is the number of features in the flock,  $\mathbf{v}$  is the mean velocity of the flock,  $\omega_f$  is a *global* color model for all flock members,  $\mathbf{W}$  is a set of  $N_f$  features,  $\mathbf{w}_i = \{\mathbf{x}_i, \omega_i\}_{i=1}^{N_f}$ , with image positions  $\mathbf{x}_i$ , and feature parameters  $\omega_i$ . Finally, the flock has a set of *collision* parameters  $\xi_c$ , and a set of *union* parameters,  $\xi_u$ . The collision parameters are used to define a function of the distance between members of the flock that indicates when two members are too close. An example is a threshold function, in which case  $\xi_c$  is a threshold on the distance. The union parameters,  $\xi_u$ , are similar, except they define when a member is straying from the flock center.

The likelihood of observing an image  $\mathbf{z}$  given a flock  $\phi$ , assumes that each feature generates parts of the image independently,

$$L(\mathbf{z}|\phi) = \prod_{i=1}^{N_f} L(\mathbf{z}|\mathbf{w}_i, \omega_f). \quad (1)$$

We use a simple type of feature, a color *speck*, which is a set of  $N_s = 4$  pixels in a  $2 \times 2$  square, with a *local* Gaussian color model,  $\omega_i = \{\mathbf{c}_i, \Sigma_i\}$ . We use a *global* Gaussian color model  $\omega_f = \{\mathbf{c}_f, \Sigma_f\}$ , and the likelihood of image  $\mathbf{z}$ , given a speck,  $\mathbf{w}_i$ , in a flock with color model  $\omega_f$ , is a product over  $N_s$  speck

pixels of two Gaussians

$$L(\mathbf{z}|\mathbf{w}_i, \boldsymbol{\omega}_f) \propto \prod_{j=1}^{N_s} e^{-\gamma_o \min(c_p, \frac{1}{2}(\mathbf{z}_j - \mathbf{c}_i)' \boldsymbol{\Sigma}_i (\mathbf{z}_j - \mathbf{c}_i))} e^{-\gamma_c \min(c_p, \frac{1}{2}(\mathbf{z}_j - \mathbf{c}_f)' \boldsymbol{\Sigma}_f (\mathbf{z}_j - \mathbf{c}_f))} \quad (2)$$

where  $\mathbf{z}_j = \mathbf{z}(\mathbf{x}_j)$  is the image color value at speck pixel  $\mathbf{x}_j$ . The  $i^{\text{th}}$  speck conforms to the flock's color model,  $\boldsymbol{\omega}_f$ , as well as to its own local color distribution through  $\boldsymbol{\omega}_i$ . Finally, a constant “background” density,  $c_p$ , gives better performance under occlusions, allowing some members of the flock to be “lost” (e.g., on an occluding object). The parameters  $\gamma_o$  and  $\gamma_c$  control the tradeoff between the local and global color models.

## 2.2 Sequential Estimation of Flock Density

This section describes how we can estimate the flock density over time using a sequential Markovian process. Let  $\boldsymbol{\phi}_t$  denote the flock at time  $t$ , and  $\mathbf{z}^t = \{z_1 \dots z_t\}$  be the observations (images) up to time  $t$ . Tracking is the estimation of the filtering distribution  $p(\boldsymbol{\phi}_t|\mathbf{z}^t)$ . This distribution is updated sequentially using the standard two-step recursion [9], in which  $p(\boldsymbol{\phi}_t|\mathbf{z}^t)$  is updated given  $p(\boldsymbol{\phi}_{t-1}|\mathbf{z}^{t-1})$  and a new measurement  $\mathbf{z}_t$ .

$$\text{predict :} \quad p(\boldsymbol{\phi}_t|\mathbf{z}^{t-1}) = \int_{\boldsymbol{\phi}_{t-1}} D(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})p(\boldsymbol{\phi}_{t-1}|\mathbf{z}^{t-1}) \quad (3)$$

$$\text{update :} \quad p(\boldsymbol{\phi}_t|\mathbf{z}^t) \propto L(\mathbf{z}_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t|\mathbf{z}^{t-1}) \quad (4)$$

where  $L(\mathbf{z}_t|\boldsymbol{\phi}_t)$  is given by Equations 1 and 2, and  $D(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})$  is the transition dynamics of a flock. There are three terms in the dynamics,

$$D(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1}) = D'_s(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1})\psi_u(\boldsymbol{\phi}_t)\psi_c(\boldsymbol{\phi}_t), \quad (5)$$

each of which describe a flocking behavior. First, due to the dynamics,  $D'_s$ , each feature moves according to the mean velocity of the flock,  $\mathbf{v}$ , but with added Gaussian noise:

$$D'_s(\boldsymbol{\phi}_t|\boldsymbol{\phi}_{t-1}) = e^{-\gamma_d \sum_{i=1}^{N_f} (\Delta \mathbf{x})' \boldsymbol{\Sigma}_v^{-1} (\Delta \mathbf{x})} \quad (6)$$

where  $N_f$  is the number of specks in the flock,  $\Delta \mathbf{x} = (\mathbf{x}_{t,i} - \mathbf{x}_{t-1,i} - \mathbf{v}_{t-1})$ ,  $\mathbf{v}_{t-1}$  is the mean velocity of the flock and  $\boldsymbol{\Sigma}_v$  is the covariance of the noise in the dynamics, assumed constant for all flock members. The third term in (5) is a penalty for being too close to another flock member, and is implemented using pairwise potentials between members of the flock, expressed as a Gibbs distribution

$$\psi_c(\boldsymbol{\phi}_t) = e^{-\gamma_c \sum_{i=1}^{N_f} \sum_{j=1}^{N_f} \delta(i \neq j) g_c(\mathbf{x}_{t,i}, \mathbf{x}_{t,j}, \boldsymbol{\xi}_c)},$$

where  $g_c(\mathbf{x}_{t,i}, \mathbf{x}_{t,j}, \boldsymbol{\xi}_c)$  is a sigmoid penalty function that varies inversely with the distance between  $\mathbf{x}_{t,i}$  and  $\mathbf{x}_{t,j}$ , with threshold  $\boldsymbol{\xi}_c$ . A similar type of penalty function was also used in [10] to model interaction penalties between different particles in a multi-target tracking example.

The second term in (5) is a penalty for being too far from the center of the flock, also implemented using a potential over the feature locations and the flock mean location:

$$\psi_u(\boldsymbol{\phi}_t) = e^{-\gamma_u \sum_{i=1}^{N_f} g_u(\mathbf{x}_{t,i}, \bar{\mathbf{x}}_t, \xi_u)},$$

where  $\bar{\mathbf{x}}_t$  is the mean position of the flock, and  $g_u(\mathbf{x}, \bar{\mathbf{x}}, \xi_u)$  is a penalty function that is proportional to the distance between  $\mathbf{x}$  and  $\bar{\mathbf{x}}$ , and can be implemented using an inverse sigmoid function.

### 2.2.1 Particle Approximation

The general recursions introduced in the previous section yield closed-form expressions only in a limited number of cases, such as when dynamics and likelihood functions are linear Gaussian. In the general case, we wish to deal with functions that may be non-linear and/or non-Gaussian, and so adopt a sequential Monte-Carlo approximation method, also known as a particle filter [9], in which the target distribution is represented using a weighted set of samples. We use a standard importance sampling method [9], with resampling in cases of degeneracy [11]. We also use a data-driven proposal distribution in order to locate a new object entering the scene, and to find an object to track again after it has been lost. The proposal distribution therefore includes two processes. The first is a commonly used dynamics proposal [12]: the expected mean flock dynamics  $D'_s$  given by Equation (6). The second is a process that generates new samples directly from a new image. The complete proposal combines these two distributions with a weight,  $\alpha$ , that gives the probability a sample will be drawn from the data-driven proposal [13].

We sample from this combined proposal by randomly sampling from one of the component distributions according to  $\alpha$ . Data samples are re-weighted using a prior weight of  $(\alpha N_p)^{-1}$ , assuming all of the expected  $\alpha N_p$  particles will be drawn equally, and a null dynamics  $D_0(\boldsymbol{\phi}_t) = D'_0(\boldsymbol{\phi}_t)\psi_u(\boldsymbol{\phi}_t)\psi_c(\boldsymbol{\phi}_t)$ .  $D_0$  includes the collision and union penalty functions, and a prior distribution,  $D'_0 = N_d^{-N_f}$ , which is the probability of drawing  $N_f$  features independently at random from a set of  $N_d$  possibilities, where  $N_d$  is the number of valid pixels used in the data proposal.

Samples are drawn from the data by first thresholding the image using a fixed set of thresholds obtained from the global Gaussian color model,  $\boldsymbol{\omega}_f$ , as:  $\mathbf{c}_f \pm \text{diag}(\boldsymbol{\Sigma}_f)$ . The resulting binary image is median filtered to remove

small components. We then remove all components that are unlikely to have arisen from the current filter, or are unlikely to represent a new object *a-priori*. Finally, we build a normalised map  $P_o(\mathbf{z}_i|\boldsymbol{\omega}_f) = kPr(\mathbf{z}_i|\boldsymbol{\omega}_f)$  where  $k$  is the normalising constant summed over the component being used.

Finally, we draw a flock sample from the joint distribution over the  $N_f$  feature locations,  $Pr(\mathbf{x}_{i=1}^{N_f}|\mathbf{z}, \boldsymbol{\theta})$ , by sampling each feature independently from  $P_o$ , giving a sample  $\phi^{(i)} = \{N_f, \mathbf{w}, \mathbf{v}_0, \boldsymbol{\xi}_{c,o}, \boldsymbol{\xi}_{u,o}\}$ , where  $\mathbf{w} = \{\mathbf{x}_i, \boldsymbol{\omega}_i\}_{i=1}^{N_f}$ ,  $\boldsymbol{\omega}_i$  are the feature parameters for the image region at  $\mathbf{x}_i$ , and  $\mathbf{v}_0$  is drawn from a zero-mean Gaussian distribution with variance  $\sigma_v$  (the expected variances of the mean velocity of appearing flocks):  $\mathbf{v}_0 \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \sigma_v \mathbf{I})$ . The flocking parameters are set to be some constant *a priori* values,  $\boldsymbol{\xi}_{c,o}, \boldsymbol{\xi}_{u,o}$ , although in general they could also be computed from the image given the sample.

Once a new set of samples has been drawn, we set the value of  $\alpha$  for the data-driven proposal by looking at the filter’s current *strength*,  $\nu$ , and whether a connected component was found that corresponds to that filter (denoted by a Boolean `comp`). The *strength*,  $\nu$ , of the particle filter is based on the total (unnormalised) weight of the prior ( $m_t^{*,p}$ ) and data ( $m_t^{*,d}$ ) particles as:

$$\nu = \max(0, \min(1, \frac{\log(m_t^*/m^-)}{\log(m^+/m^-)})) \times 10, \quad (7)$$

where  $m_t^* = m_t^{*,d} + m_t^{*,p}$ ,  $m^+$  and  $m^-$  are the maximum and minimum weights, respectively, and the weights are clamped to lie between  $[0, 1]$ . If no component was found, then  $\alpha = 0$ , since there will be nothing to draw samples from anyways. Otherwise, we set  $\alpha$  closer to 0 the higher the strength. If  $\nu > 8$ , then we set  $\alpha = 0$ . In general,

$$\alpha = (\text{comp} > 0)(1 - \min(1, \nu/8)). \quad (8)$$

### 2.3 Multiple Objects

In our application domain, we wish to track a number of objects simultaneously, such as the hands and the towel. Since the number of objects will be bounded above by a small number (e.g., three), we can simply use independent filtering processes. Each particle filter operates exactly as described above, but if multiple objects have the same global models,  $\boldsymbol{\omega}_f$ , then we will have to slightly modify our assignment procedure for building the data-driven proposals. The idea is that, given a number of connected components in the image, we want to assign components to filtering processes in a suitable way. We do so by computing the distance from each connected component to each filter mean, and adding a small penalty for filter strength, to allow the weaker

```

initialise all particles evenly and  $t = 0$ , then repeat
1. get new image  $\mathbf{z}_{t+1}$ 
2. smooth image and convert to HSV color space
   if any filter's strength is small
3.     threshold HSV color and get connected components
4.     assign components to filters
       while two filters assigned to same component:
5.         break the component in two using K-means
6.         assign components to filters
7.     set  $\alpha$  values for all filters (Eqn 8)
8.     construct data-driven proposal samples for all
           filters with  $\alpha > 0$  (Sec. 2.2.1)
   else
9.     set  $\alpha$  to 0 for all filters
   end if
10. update all filters, compute new strengths (Eqn. 7)
13.  $t \leftarrow t + 1$ 

```

Fig. 3. Sequential Monte-Carlo algorithm

filters more access to new data, all else being equal. Components are then assigned to their closest filter. In some cases, two filters will be assigned to the same component. For example, when the hands are together during rinsing or applying soap, they appear as a single connected component. In such cases, we use a K-means procedure, initialised with the current filter means, to break the large components into two pieces, and then re-assign. We also ensure this happens if the two hand filters are strong and very close together (less than 30 pixels apart), as may sometimes happen if the two hand filters get locked onto the same hand. Finally, we add an *interaction* penalty to the likelihood function that penalises the two hand trackers for being too close together. When the hands are touching one another, we track this with a single tracker, the other tracker will have a low strength and will be encouraged to look for something else to track. An outline of the algorithm is in Figure 3.

### 3 POMDP model

A discrete-time POMDP consists of: a finite set  $S$  of states; a finite set  $A$  of actions; a stochastic transition model  $\Pr : S \times A \rightarrow \Delta(S)$ , with  $\Pr(t|s, a)$  denoting the probability of moving from state  $s$  to  $t$  when action  $a$  is taken, and  $\Delta(S)$  is a distribution over  $S$ ; a finite observation set  $O$ ; a stochastic observation model with  $\Pr(o|s)$  denoting the probability of making observation  $o$  while the system is in state  $s$ ; and a reward assigning  $R(s, a, t)$  to state transition  $s$  to  $t$  induced by action  $a$ . Intuitively, the system actions cause stochastic

state transitions, with different transitions being more or less rewarding (reflecting the relative desirability of the states, the costs of taking actions, or some combination). States cannot be observed exactly: instead the stochastic observation model relates observable signals to the underlying state. The POMDP can be used to monitor beliefs about the system state using standard Bayesian tracking/filtering. A variety of techniques can be used to compute a *policy* that maximizes (or approximately maximizes) the expected discounted sum of rewards. Since the system state is not known with certainty, a policy maps *belief states* (i.e., distributions over  $S$ ) into choices of actions. We refer to [14] for an overview of POMDPs.

The model we currently use is specified manually, using prior knowledge of the domain. The model was built iteratively on two previous versions, the first a fully observable (MDP) [15] and the second a POMDP evaluated only in simulation [2].

### 3.1 Handwashing States

The handwashing task is modelled as a POMDP with nine state variables, three observation variables, and 25 actions. There are 207360 states and 198 observations. The set of state variables can be divided into three groups: *task*, *attitude* and *bookkeeping* variables. We use the term *sequence* to denote a single handwashing event, and *trial* to denote a set of handwashing sequences, possibly on different days over the course of many weeks. Figure 4 shows the POMDP as a Dynamic Bayesian network. Only *task* and *attitude* variables are shown, as well as observations, actions, and rewards.

The *task* is described by two variables, *planstep* (**PS**) and *behavior* (**BE**). The *plansteps* break the handwashing task down into eight situations, each a characterization of the state of the task: are the hands dirty, soapy or clean; are the hands wet or dry; is the water on or off. The plansteps are depicted as nodes in the *planstep graph* in Figure 5. User *behaviors* cause transitions in the *planstep graph* as shown in Figure 5, and correspond to one of six activities: using soap, at water, at tap, at sink, drying, or away. Whereas the planstep is sufficient to characterise the state of the hands, *behavior* is needed to fully monitor the progress. For example, a user can be in planstep **a** with hands at the soap (trying to get to planstep **e**) or at the tap (trying to get to **b**).

The user *attitude* has three factors: *dementia\_level* (**DL**) = {*low, med, high*}, giving the user’s overall level of dementia (low dementia means more demented), *awareness* (**AW**) = {*never, no, yes*}, telling whether the user is aware of what they are doing, and *responsiveness* (**RE**) = {*none, max, med, min*}, giving what type of prompts the user is responsive to. We assume that *de-*

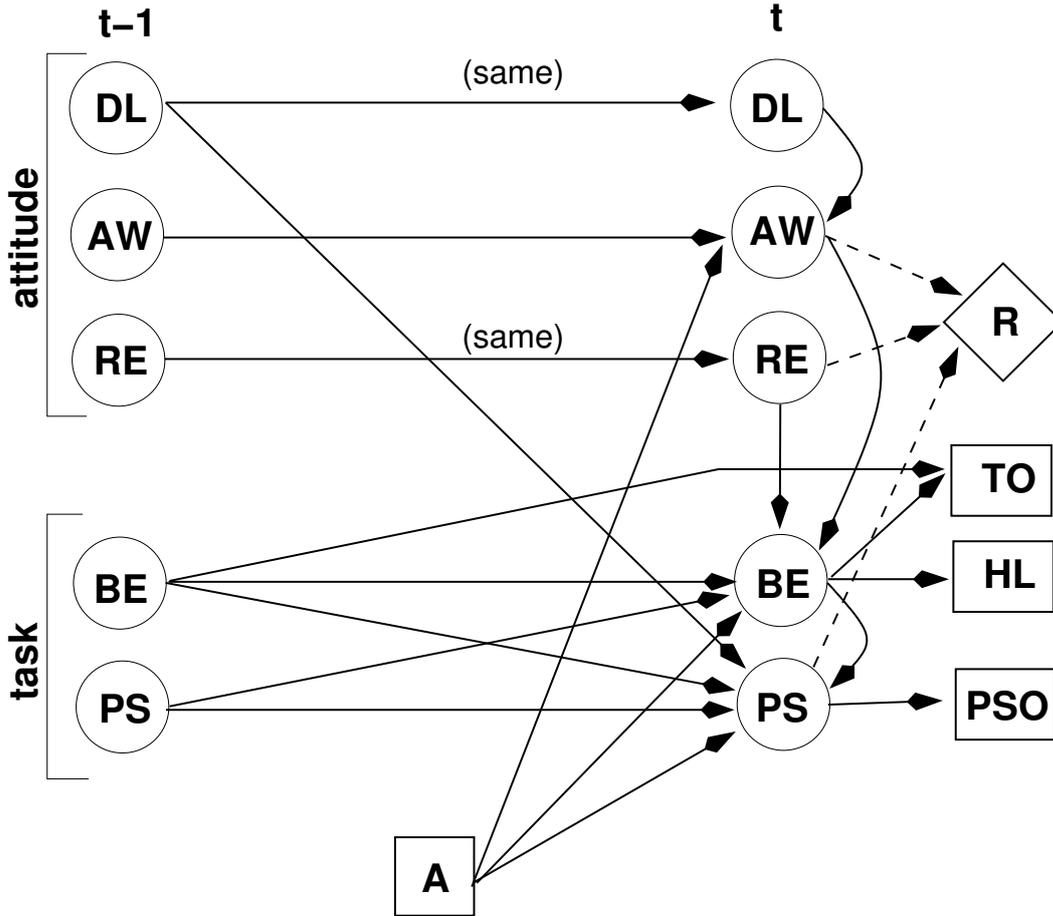


Fig. 4. Dynamic Bayesian network representation of the POMDP model, showing only a subset of variables (the *task* and *attitude*). Observations (rectangles on right) are shown only from time  $t$ , but are replicated across all times. The actions are  $A$ , and are prompts or caregiver calls. The diamond  $R$  is the reward function, with dependencies shown as dashed lines. The variables are **DL**: *dementia level*, **AW**: *awareness*, **RE**: *responsiveness*, **BE**: *behaviour*, **PS**: *planstep*. The observations are **TO**: *timeout*, **HL**: *handlocations*, **PSO**: *planstep observation*.

*dementia\_level* does not change over a clinical trial (eight weeks). However, as we discuss in Section 3.1, and as we show in our results, our model can be used to *estimate* a particular user’s level of dementia over the course of a clinical trial. A user’s *responsiveness* does not change over the course of a single handwashing sequence, but can change from sequence to sequence (e.g., from day to day). However, the POMDP beliefs can change over the course of a sequence. Finally, the user’s *awareness* can change during a sequence, depending on a number of factors, such as whether she is given prompts, whether a human caregiver intervenes, and also her *dementia\_level*. If *dementia\_level* is low, the user is less likely to gain awareness. If *dementia\_level* is high (less demented), the user is more likely to become and remain aware.

Finally, two book-keeping variables are used. The first, *last\_prompt*, ensures

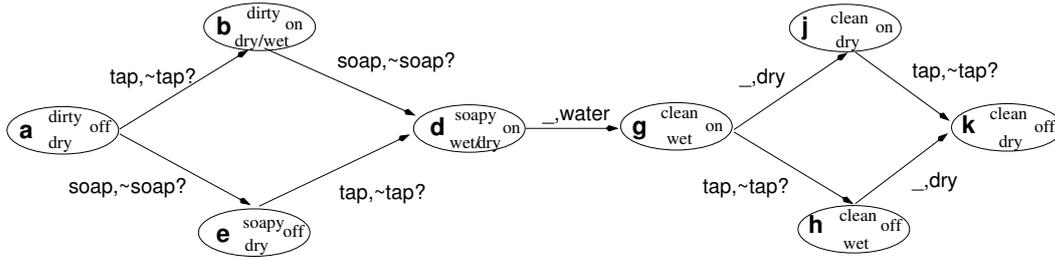


Fig. 5. Simplified view of the planstep transitions for the handwashing problem. The plansteps are shown along with the state of the hands (dirty,soapy,clean), the wetness of the hands (dry,wet) and the water flow (on/off). Transitions are for pairs of pre/post action behaviors for the null action. Transitions for prompting actions are slightly different (see text). The model is simplified for presentation in that some additional transitions and regressions are possible. An underscore (  $\_$  ) means any behavior and  $\sim b$  means any behavior other than  $b$ . A question mark,  $?$  indicates a probabilistic transition.

that the effects of the prompts persist for more than one time interval. This allows for the user to respond to the prompt after two time intervals. The other book-keeping variable, *hands\_washed* gives whether the task is complete or not. This is used for the reward function.

### 3.2 Handwashing System Actions

category	transitions	prompt	effect
wet hands	$d \rightarrow g$	<i>wet_hands</i>	<i>put hands under water to rinse</i>
water on	$a \rightarrow b, e \rightarrow d$	<i>turn_on_water</i> <i>tap_on</i>	<i>move hands to taps</i> <i>push up on the tap to turn on</i>
water off	$g \rightarrow h, j \rightarrow k$	<i>turn_off_water</i> <i>tap_off</i>	<i>move hands to taps</i> <i>push down on the tap to turn off</i>
soap	$a \rightarrow e, b \rightarrow d$	<i>use_soap</i> <i>use_pump</i>	<i>move hands to soap</i> <i>push down on soap to apply</i>
dry hands	$g \rightarrow j, h \rightarrow k$	<i>dry_hands</i>	<i>take towel and dry hands</i>
caregiver	any	<i>caregiver</i>	<i>human assistance: next planstep</i>

Table 1

Prompt categories showing the transitions in the plansteps intended by the prompts (see Figure 5), the different prompts for each category and what the prompt is meant to help the user do.

There are three types of action the system can take: to do nothing, to prompt, or to call for human assistance. The prompts come in five categories, corre-

sponding to the transitions shown in Figure 5, as shown in Table 1. In each category, there are one or two prompts. In the case of turning the taps on or off, or using the soap, there are two prompts. The first is meant to advance the user to the object in question (taps or soap), the second to help them finish that step. Each prompt comes in three levels of specificity: *minimal*, *maximal*, and *video*, giving the level of detail in the prompt. Each prompt is preceded with a reminder that the user is washing their hands. A minimally specific prompt gives a basic verbal instruction (e.g., “Use the soap now”). A maximally specific prompt adds the user’s name, and more detail (e.g., “John, use the soap on your left in the pink bottle”). The video (most specific) prompts add a video demonstration to the maximally specific prompt’s audio track.

### 3.3 Observations and observation function

There are three observation variables in the model. The planstep observation (**PSO**) is the caregiver’s indication of the planstep after an intervention. The system must gain information after an intervention to avoid repeatedly calling for assistance. The second observation is a timeout (**TO**) that indicates that the user has not changed behavior for a fixed (long) interval. The hand location observation (**HL**) comes from the hand tracker as described in Section 2, and gives the current locations of the hands and towel. The mean positions of the three tracks are spatially discretised into a coarse and pre-defined set of areas using threshold distances to each object (e.g., taps, soap), as shown in Figure 6, and combined to form the values for **HL**. Examples include *both\_at\_soap* and *water\_towel* meaning one is at water and one at towel. These observations are conditioned on the *behavior* through the probability distribution  $P(\text{hand\_locs}|\text{behavior})$ . We have also investigated methods

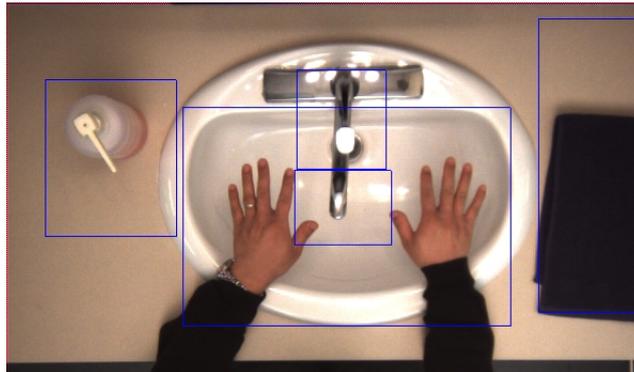


Fig. 6. Handwashing regions for observation discretisation. These regions have to be manually specified for each sink-soap configuration. The towel region is not used if the towel is being tracked.

for learning this discretisation automatically using top-down value-directed

information from the POMDP, but have not fully integrated these techniques into the working system yet [16].

### 3.4 Dynamics

Mapping the actions of the hand tracker to the state of the POMDP requires a form of temporal abstraction. The hand tracker operates at 10-20 frames per second and a new hand location estimate can be made every 50-100 ms. However, it is not appropriate to update the state of the POMDP that quickly. One of the main reasons for this difference is the time scale at which prompting actions themselves occur. A typical prompt (e.g., “use the soap now, John”), takes about a second or more to state (depending on its level of specificity). We take this to be a rough estimate of the time scale at which the POMDP should operate. This temporal abstraction step can be integrated into the model directly by adding explicit temporal segmentation variables that can be treated as random variables in the dynamic Bayesian network [17]. However, this imposes an unacceptably large computational burden on POMDP solution techniques by causing an unnecessary blow up in the size of the state space (this holds even when performing belief state monitoring/inference alone). Therefore, we adopted a simple heuristic to decide when the policy should be consulted in the POMDP, and thus when the belief state should be updated.

Our heuristic approach updates the belief state in one of two situations. First, when the hand positions have changed and this change will cause a significant change in the belief state. At each belief state update, the reported hand positions are recorded. At each subsequent frame, the hand positions are compared to those recorded at the previous belief update. If there is any change, then a belief update is simulated, and the resulting, new belief is compared with the previous one using the Kullback-Leibler divergence. If the belief is going to change significantly, then this new belief is tested over a fixed number of frames to make sure that it persists. The fixed number changes based on the planstep, allowing some behaviors to be recognised more quickly. A persistent, new belief triggers a real belief update for the system, a policy lookup, and possibly a system action (prompt).

The second cause for a belief state update is if the person has not changed her behavior (e.g., has not moved) for a long period of time, termed a *timeout*. These explicit timeouts are the third observation, and are an indication that the user is not aware. In the POMDP, we condition the timeout observation on the joint pre- and post-action behaviors being identical. Since the behaviors are conditioned on the awareness, a timeout will give evidence for lower awareness.

There are two important factors in the dynamics of the POMDP. The *behavior*

*dynamics* is the distribution over behaviors given the current state and the action taken by the system. These reflect the probability that the user will behave (or respond) appropriately given a particular system prompt (or no prompting at all) and the user’s responsiveness and awareness levels. The *planstep dynamics* characterize the distribution over updated plansteps given the previous planstep, an observed behavior pair (pre- and post-action) and the system action taken. These reflect the fact that behaviors cause planstep transitions (see Figure 5). Some transitions (marked with a ? in Figure 5) include the probability that the user has abandoned her attempt to complete the step. For example, if the user is in planstep **a**, and moves her hands from the taps to the sink, then the probability that she turned the water on may be less than one—she may have abandoned the step of turning on the water. Specific values can be set for particular users and tap/soap configurations.

The transitions shown in Figure 5 are only for the **do nothing** action (no prompt). Prompting actions essentially remove transitions that cannot be inferred after a prompt since the user may be responding to a prompt instead of changing behaviors due to a step completion. For example, suppose the user is in planstep **a**, and that their hands are currently at the tap (they are trying to get the water on). If they are observed to be returning from the tap on their own volition (no prompt), we can assign some probability, say  $q$ , to the water having been turned on, based on the user’s characteristics and the tap configuration. If, on the other hand, they are observed returning from the tap as a response to a prompt to use the soap, then we can no longer assign  $q$  to the water being on, because the prompt makes it much more likely that they abandoned the task in response to the new request to do something different.

### 3.5 Estimating User Attitude

The POMDP model also estimates a particular user’s *attitude* over time. In particular, the model can estimate a user’s level of dementia by watching her long-term handwashing behavior over multiple sequences. The ability to estimate user traits allows the model to report such findings to carers, and can also give the model information that can be leveraged in the policy. When a new user starts using the system, the *dementia\_level* variable has some prior distribution set based on the population of users. Over the course of each handwashing sequence, this distribution will shift slightly. If we then propagate this information from sequence to sequence (this is the only variable whose information is propagated), then we get a long-term estimate of the user’s *dementia\_level*. We show examples of this in Section 5.

It is important to note that the model’s estimate of the *dementia\_level* variable is not a clinical evaluation of the user’s level of dementia, but rather

only a functional measure of how independently a user can wash their hands. In this study, we investigate whether any correlation can be noted between this functional measure and the Mini-Mental State Examination (MMSE) results [18]. Given the small quantity of data, we can only make some qualitative evaluations, leaving more quantitative measures to future clinical trials.

### 3.6 Rewards

The reward is a combination of three main factors. First, a large reward for task completion (*planstep*  $k$ ). Second, prompts are costly, and more specific prompts cost more, due to the inducement of feelings of reduced independence in the user. Prompts are also more costly if the user is more aware. Caregiver calls are expensive if the user is aware or responsive. Third, prompting twice in a row incurs an additional cost.

### 3.7 Policy Optimization

In this section, we describe how a policy is computed for a specific POMDP model. The size of our model puts it well beyond the reach of any exact solution techniques. We used the SymbolicPerseus package [19]<sup>1</sup>. It implements a point-based approximate solution technique based on the Perseus algorithm [20] combined with Algebraic Decision Diagrams [21] as the underlying data structure. As mentioned earlier, a policy for a POMDP model consists of a mapping from beliefs (e.g., distributions over states) to actions. We denote a belief by  $b$  and a policy by  $\pi$ . Assuming an infinite planning horizon and a discount factor (a number  $\gamma$  between 0 and 1 indicating by how much future rewards are reduced), the value  $V^\pi(b)$  of a policy for some belief  $b$  may be written recursively as follows:

$$V^\pi(b) = E_b^\pi(R) + \gamma \sum_o \Pr(o|b, \pi(b)) V^\pi(b_{\pi(b),o})$$

where  $\Pr(o|b, a) = \sum_s b(s) \sum_{s'} \Pr(s'|s, a) \Pr(o|s')$ ,  $b_{a,o}$  denotes the belief reached from  $b$  after executing  $a$  and observing  $o$ , and  $\pi(b) = a$  is the action that policy  $\pi$  maps  $b$  into. Given a POMDP model, an optimal policy  $\pi^*$  is a policy that maximizes the expected sum of discounted rewards (i.e.,  $V^{\pi^*}(b) \geq V^\pi(b) \forall b, \pi$ ). Note that this value function gives an estimate of the value of being in a particular belief state, and is used to compute the best possible

<sup>1</sup> code available at <http://www.cs.uwaterloo.ca/~ppoupart/software>

action, and not as a direct performance evaluation measure when the system is in use.

The SymbolicPerseus package uses dynamic programming to compute the optimal value function  $V_n$  for  $n$  steps-to-go from the optimal value function  $V_{n-1}$  for  $n - 1$  steps-to-go:

$$V_n(b) = \max_a E_b^a(R) + \gamma \sum_o \Pr(o|b, a) V_{n-1}(b_{a,o}) \quad (9)$$

However, since  $b$  is continuous, the above computation will only be feasible if  $V_n(b)$  has a closed form parameterization. Sondik and Smallood [22] showed that the optimal value function for finite horizons is piecewise linear and convex. This means that there exists a set  $\Gamma_n$  of linear segments  $\alpha$  such that  $V_n(b) = \max_{\alpha \in \Gamma_n} \alpha(b)$ . Here  $\alpha$  can be represented by a vector indexed by states such that  $\alpha(b) = \sum_s b(s) \alpha(s)$ . Since the set  $\Gamma_n$  often grows exponentially with the horizon, it is common practise to approximate  $V_n$  with a subset of  $\Gamma_n$ . The class of point-based value iteration algorithms achieves this by selecting a set of beliefs  $B$  and computing for each  $b \in B$  the  $\alpha$  that has the highest value at  $b$ . This is done by decomposing Equation 9 in three steps. Given a belief  $b$  and the set  $\Gamma_{n-1}$  representing  $V_{n-1}$ , we can compute the best  $\alpha_b \in \Gamma_n$  at  $b$  (and hence  $V_n$ ) as follows:

- (1)  $\alpha_{b,a,o} = \arg \max_{\alpha \in \Gamma_{n-1}} \alpha(b_{a,o})$  for all  $a, o$
- (2)  $a_b = \arg \max_a E_b^a(R) + \gamma \sum_o \Pr(o|b, a) \alpha_{b,a,o}(b)$
- (3)  $\alpha_b = E_b^{a_b}(R) + \gamma \sum_o \Pr(o|b, a_b) \alpha_{b,a_b,o}(b_{a_b,o})$

Among the class of point-based value iteration algorithms, Perseus [20] is very simple and yet quite effective. The algorithm selects a set  $B$  of reachable beliefs by executing a default (or random) policy for a number of steps. Then, at each iteration it computes a set  $\Gamma_n$  by selecting beliefs in a randomized order and performing a point-based backup for each belief that doesn't already have an  $\alpha$ -vector in  $\Gamma_n$  that improves on the best  $\alpha \in \Gamma_{n-1}$  at  $b$ .

Symbolic Perseus improves on Perseus by representing  $\alpha$ -vectors *symbolically* with Algebraic Decision Diagrams (ADDs) [21] instead of by enumerating the values for each state. The transition probabilities, rewards and  $\alpha$ -vectors are represented with decision trees that aggregate identical values and allow branches to merge, potentially reducing memory and computation by an exponential factor. This idea was used successfully for MDPs [23] and for POMDPs [19]. SymbolicPerseus makes three additional approximations. First, we put a cap on the complexity of  $V_n$  (e.g., size of  $\Gamma_n$ ). This is done by retaining only the  $k$   $\alpha$ -vectors that improve the most the value function at some belief in  $B$ . Typically, this bound only causes minimal decrease in the quality of a solution. Second, we merge states with values that differ by less than the Bellman

error [24]. This allows the resulting algebraic decision diagrams to be more compact. Note that the error introduced shrinks to zero as the computation converges, preserving optimality. Third, we only sum over observations with a non-negligible probability of occurrence in Steps 2 and 3 above. More precisely  $\Pr(o|b, a)$  is negligible if  $\Pr(o|b, a)(R_{\max} - R_{\min})/(1 - \gamma) < \text{bellmanError}$ , which again preserves optimality. This fast technique is related to a general method for dealing with large observation spaces [16].

We solved the POMDP with SymbolicPerseus by using 150 alpha vectors and 65 iterations in 42 hours on a dual Intel® 2.40GHz XEON™ CPU with 4Gb of RAM, using about 2Gb of memory maximum.

## 4 Implementation and Experimental Methods

This section presents the three experiments we have done. The first (Section 4.1) evaluates the performance of the tracking system, the second (Section 4.2) evaluates the performance of the decision making component (the POMDP policy) and the third (Section 4.3) evaluates a full working system during an eight-week user trial conducted with six persons with moderate-to-severe dementia. Results from these three experiments are presented in Sections 5.1, 5.2, and 5.3, respectively.

### 4.1 Tracker Evaluation Method

The performance of the hand-and-towel tracking system is evaluated on three handwashing sequences by manually checking each frame in the sequences for tracker failures, and comparing against a simple hand-position estimator using only color (no tracking). We show how the trackers are robust over long periods of time, and can recover from failures.

### 4.2 POMDP Evaluation Methods

Simulations were used to evaluate the POMDP policy with respect to the POMDP model in simulation and show that the policy is reasonable. This demonstrates that, if the POMDP is an accurate model of the real world, then we expect our policy to perform fairly well.

Simulations use two models: the first is the true user model, and the second is the system’s model. The true user model has a fixed user *attitude*: dementia level, responsiveness and awareness. The system’s model then interacts with

the true model, and attempts to estimate the state of the user and take actions accordingly. We evaluate the simulations by looking at the average discounted reward over time. We compare this reward to that obtained by the optimal policy for the fully observable version of the true user model, solved using a fast, structured, factored value iteration method [23]. This fully observable version has access to the actual state of the true user model, and so represents a utopian upper bound on the value that can be achieved. Note that this upper bound does not take any observational uncertainty into account and is therefore higher than the value that would be obtained by the optimal POMDP policy. Also, as discussed in Section 3.5, we look at whether the estimated *dementia\_level* converges to the *dementia\_level* of the true user model.

We compare five different policies. The first is the computed POMDP policy, while the other four are heuristic policies. The first heuristic has a fixed set of hand-crafted thresholds on the belief distribution, and attempts to prompt when the user is not aware, and does so at whatever level of responsiveness is most likely. If the user is unaware and unresponsive, then the human caregiver is called. The other heuristic policies that we compare against are the Null policy, which always does nothing, the CG policy, which always calls the caregiver, and the certainty-equivalent (CE) policy, which looks at the most likely state given the current belief, and then acts according to the policy derived for the fully observable model (MDP).

We compute two different estimates of the discounted rewards obtained in simulation averaged over all 36 possible attitudes (values of *dementia\_level*, *awareness* and *responsiveness*). The first,  $\rho_0$  is the expected discounted reward given a uniform distribution over user attitude. The second,  $\rho_\delta$  is the expected discounted reward using the initial belief distribution over attitudes, which gives a more accurate (slightly lower) estimate since the actual user population will consist of more persons with lower dementia levels. Standard deviations of the averages are computed using the differences between the POMDP policy rewards and that of the fully observable (utopian) MDP upper bound.

### 4.3 User Trial Methods

A modified withdrawal-type, single-subject research design was used to test the system. This research design consisted of a baseline phase (no computer guidance), *A*, and an intervention phase (with computer guidance), *B*, tested in the order *A1* – *B1* – *A2* – *B2*. For each test phase, data were collected describing the ability of the participants to independently perform the required task (handwashing), and caregiver burden and workload. Data were also collected on the performance of the system itself, including its accuracy in determining the type of assistance required by the users, and the effects of

the chosen system response.

Participants were recruited from a long-term care facility in Toronto, Canada, where the trials were conducted. The primary inclusion criteria were clinical diagnosis of moderate-to-severe dementia (from our experiences this group requires, and benefits from, cuing during ADL completion). Six older adults participated in the study—5 females, 1 male, average age  $86.3 \pm 8.8$ . Using the Mini-Mental State Examination [18], five of the subjects were classified as having moderate-level dementia, with the one remaining classified as severe.

As shown in Figure 8, a fully functional washroom located in the long-term care unit was retrofitted with the necessary system hardware, specifically a ceiling-mounted IEEE-1394 digital video camera (Point Grey Research DragonFly2), a 21 inch LCD screen and speakers (to play/display the required prompts), and a Dell Latitude laptop computer (2 GHz processor, 2 Gb RAM) as the processing unit for the system software and hardware. This computer also served as the primary graphical user interface (Figure 7), displaying information about the system variables (e.g., estimated plan steps, response provided), and the user’s progress through the task, and allowing the human operator to input observations during caregiver calls (details in Section 4.3.1). The laptop was located in an office adjacent to the test washroom, where the researchers monitored the trials and system operation. The trials were also recorded using an independent video recorder from a view above and slightly behind the sink (see Figure 8). This video is not used by the system, but is used for the manual annotations. All system actions were logged, including the flock particles, actions, times stamps, and images.

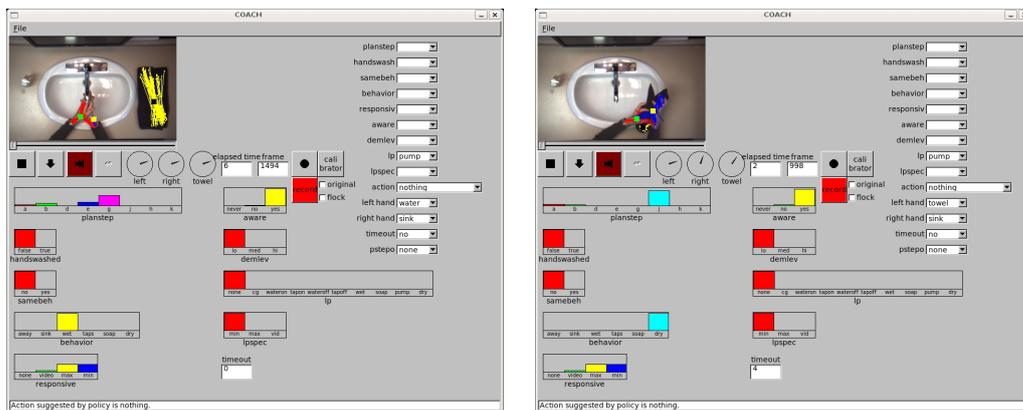


Fig. 7. Graphical User Interface (GUI) for monitoring user and system progress during the trials. Three screenshots are shown during a trial. The system video showing the flock trackers is in the top left. The POMDP belief state is shown as a set of bar plots in the bottom left quadrant. The pull-down menus along the right side are for entering state information to the system after a caregiver call.

### 4.3.1 Trial Methodology

Handwashing sequences involved two assistants. The first assistant is the caregiver, whose role was to assist the user either when instructed to do so by the system (during intervention phases) or when she thought it was necessary (during baseline phases). The second assistant is the technician, whose role was to start/stop the system, and to monitor the system running through the interface (Figure 7). Each subject was required to sit in a wheelchair before being positioned by the caregiver in front of the sink within the test washroom and asked to complete the task of hand washing, at which point the device was started by the technician. During the intervention phases the caregiver was instructed to leave the subject alone inside the test washroom and to join the technician in the adjacent room. The caregiver provided a subject with assistance only after an audible signal from the device’s GUI directed her to do so, at which point the system would pause. The caregiver would then assist the subject with that particular step only and return to adjacent room. The technician would then input the current planstep observation, restart the system, and the sequence proceeded. Finally, the caregiver was instructed to veto the system in case of any circumstance where the person’s health or well-being was being jeopardized, or if they started to move away from the sink (e.g., attempting to leave the room). This occurred in only two cases, which were removed from the data discussed below. Each subject’s sequence took approximately twenty minutes to complete, which included time to accompany him or her to and from his or her room. The caregiver and technician wrote logs of the sequences, documenting any special events for each day.

The user’s dementia level was estimated using the Mini-Mental State Examination [18] before and after the POMDP trials. The Mini-Mental State Examination (MMSE) is a standardized written assessment instrument that measures and evaluates cognitive function and mental impairment, often given serially to determine the current abilities and disabilities of an older adult from the effects of time. The MMSE includes a series of thirty questions with stimuli that measure comprehension, reading, writing, memory, and drawing, each of which scores points if answered correctly. If every answer is correct, a maximum score of 30 points is possible. A user’s score classified him or her into one of three categories: (1) mild 20-25; (2) moderate 10-19; (3) severe 0-9.

### 4.3.2 Annotation Methodology

The videos and logs were manually annotated for the unobservable aspects of the POMDP state (planstep, behavior, awareness and responsiveness). Annotations of the data were made by a single human rater using the utility shown in Figure 8. The screen depicting the trial video would pause at the moments in the sequence when the POMDP updates were made. At each pause, the

human rater would annotate the subject’s behavior, hand locations (left and right), attitude and the planstep. The rater was briefed on the different hand locations and behaviors prior to starting. An additional behavior of N/A was available if the annotator could not decide what behavior was being executed. The variables *awareness* and *responsiveness* were annotated using sliders with

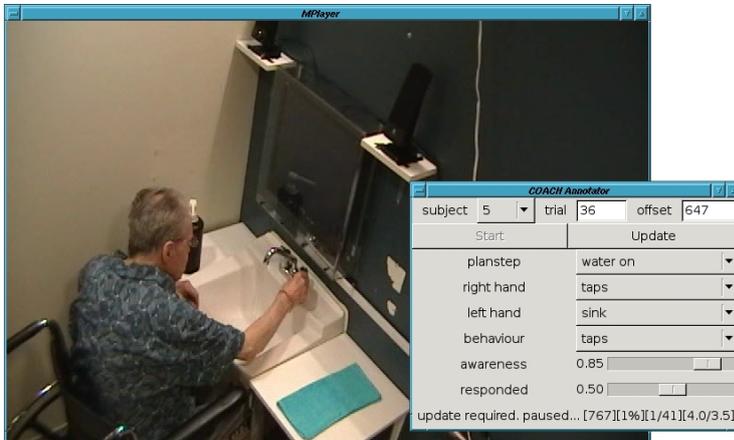


Fig. 8. Example during handwashing sequence annotation showing played video and annotator utility GUI. The picture is taken from an independent video camera (not used by the system), and shows the sink, soap, towel, a user, the video display behind plexiglass and two speakers mounted above the display. The system camera is out of sight, above the sink, looking down.

the range  $[0-1]$  in 0.01 unit increments. Higher values on the *awareness* scale reflected stronger belief that the subject would not need any assistance (prompt) from the device. Higher values on the *responsiveness* scale reflected a higher level of confidence by the rater that the subject had been responsive to any correct or incorrect assistance (prompt) they may have received. The slider associated with the variable *awareness* was updated at every pause. In contrast, the *responsiveness* variable was only updated after a complete sequence, at which point the human rater input whether she believed the subject was responsive or not to any assistance or prompts given by the COACH (0.5 was given if no prompts were issued). This responsiveness annotation was then used for all steps during the sequence. It is important to note that currently there is only one manual annotation, and so the annotations represent only one human’s (possibly biased) estimates. We are currently working on more annotations, and will include these in future versions of the paper.

#### 4.3.3 Performance Evaluation

We evaluate the performance of the system by looking at examples, at the correlation between POMDP estimates of the behaviors and manual annotations of the behaviors, at the reward gathered, and at the correlation between MMSE scores and the POMDP’s estimated dementia levels.

The correlation between the manual annotations and the POMDP belief estimate how well the computer vision system is able to predict the manual observations of behaviors. This gives us some information about the computer vision system’s interface to the POMDP model. However, note that better correlation here does not necessarily indicate better performance for the POMDP, because the strength of the POMDP model is to estimate the entire *distribution* over states. Thus, we do not look at other correlations (e.g. between annotated plansteps and POMDP estimates).

The expected reward, based on the POMDP belief state, is

$$V(b_0) = \frac{1}{N} \sum_{j \in \text{sequences}} \sum_{i \in \text{steps}} \sum_{s \in \text{states}} \gamma^i b_{ji}(s) R(s, a_{ji}), \quad (10)$$

where  $N$  is the number of sequences,  $\gamma$  is the discount factor,  $b_{ji}(s)$  and  $a_{ji}$  are the POMDP belief and action taken, respectively, at step  $i$  of sequence  $j$ , and  $R(s)$  is the reward function for state  $s$ . This estimate gives what the POMDP believes its reward to be, and may be an over or under-estimate of the actual reward. The actual reward can be computed based on the manual annotations as

$$V(s^*) = \frac{1}{N} \sum_{j \in \text{sequences}} \sum_{i \in \text{steps}} \gamma^i R(s_{ji}^*, a_{ji}), \quad (11)$$

where  $s_{ji}^*$  is the manual annotation of step  $i$  in sequence  $j$ . This is our best estimate of the actual value of the POMDP’s actions, since it is based on independent manual annotations of each sequence. The expectations can be compared to the POMDP simulation results described in the previous section (4.2). These simulation results give the expected reward for a POMDP acting in a world that is simulated using the POMDP model itself (so the POMDP has a perfect model of the world dynamics and observation distributions). On the other hand,  $V(s^*)$  gives the reward for a POMDP acting in the real world, and so the difference gives us an indication of how well our POMDP model approximates the real world.

The correlation between the MMSE scores and the POMDP’s estimated dementia levels is a first attempt to validate the use of handwashing as a method to estimate a user’s overall dementia level. The system was designed to have the belief in the *dementia\_level* variable propagated across trials for each user, to see if it would converge to a value that correlated with the MMSE scores, as described in Section 3.5. In this experiment, this belief propagation was done offline after the trials, by simulating the POMDP belief state using the actual logged data, including the actual system actions that were performed during the trials. This method is sound, and gives the true belief in the *dementia\_level* that would have been obtained using the actions that were performed by the system during the actual trial. We use this as a proof of this belief propagation

concept, in preparation for using it in real-time during future user trials.

## 5 Results

### 5.1 Tracker Evaluation Results

We used sequences taken from a user trial (completed in 2006) in which an automated prompting system (a predecessor of the one reported here) monitored and prompted persons with moderate to severe Alzheimer’s disease [25]. The video was taken from an overhead SONY CCD DC393 color video camera at  $30fps$ , and a  $570 \times 290$  pixel region around the sink was cropped. We used 200 particles and could perform updates of all three filters at over 13 frames per second. We evaluated the tracker by looking at whether the mean flock position was inside the hands or towel region in each frame for 1300 frames from a single user’s sequence during which the user was drying their hands. We compare our method to a simple heuristic that looks only at the connected components from the thresholded images (using  $\theta_f$ ). We find our method makes no errors (0%) in locating the towel during the extreme occlusions compared to 7.4% for the heuristic method. The error rates for hand locations were 2.4% for our method vs. 5.3% for the heuristic method. The errors for our method in locating the hands were due to one hand’s flock migrating close to the other hand when the hands were close.

We also tested our method on six sequences from two different users, and measured the number of tracker failures. We only looked at frames in which both hands were present and at least one was partially visible, and in which the caregiver was not present. A tracker failure was noted either if the hands were separated but one was not tracked, or if both hands were present and together (e.g., when being rubbed together) but neither hand was tracked, or if the towel was not tracked. We found error rates of only 1.9% over a total of 16986 frames in 3 sequences for one user and 0.8% over a total of 7285 frames in 3 sequences for the other. The majority of tracker failures happened after an abrupt change in hand motion, due to our constant velocity assumption. The tracker was consistently able to recover after all tracker failures within about 10 frames.

Figure 9 show a close-up of an example where the data-driven proposal is used. The hands are being tracked in the sink region at frame 2144. The strength of the right hand filter is weak (6), since the hand has begun to move forward, and the particles do not adjust to the new velocity fast enough. The corresponding value of  $\alpha$  is 0.3 at frame 2145, meaning that about a third of the particles at the next frame will be drawn from the data-driven proposal,

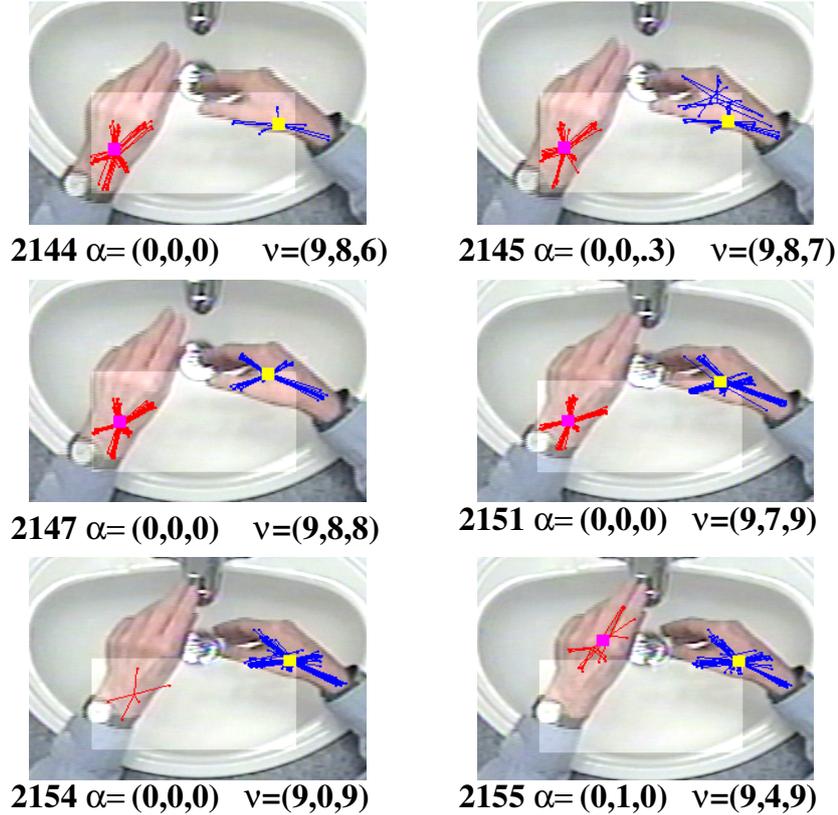


Fig. 9. Example sequence showing the data-driven proposal being used for the right hand (at frame 2145) and showing a complete reset (with  $\alpha = 1$ ) for the left hand at frame 2155. The  $\alpha$  values and strengths  $\nu$  are shown for (towel, left hand, right hand). The lightly colored region shows that both hands are estimated to be at the *sink* region.

$q_d$ , with the remaining coming from the prior dynamics. Indeed, we see the newly proposed particles at frame 2145, well distributed across the right hand. However, the weights of these particles, although sufficient to remain in the filter, do not significantly affect the mean of the filter immediately. By frame 2147, these new particles have gained all the filter’s weight, however, and the hand is well tracked again through frame 2155, where the left hand filter is losing strength. At frame 2154, the left hand filter has dropped to 0 in strength, and gets completely re-initialised by the data-driven proposal at frame 2155.

## 5.2 POMDP Evaluation Results

A simulation experiment involved a set of 20 simulations of handwashing, each for 100 steps (or until the true user had her hands washed). The dementia level belief was propagated across the 20 simulations, and we did 10 experiments with different random seeds. Table 2 shows the expected rewards over the 10

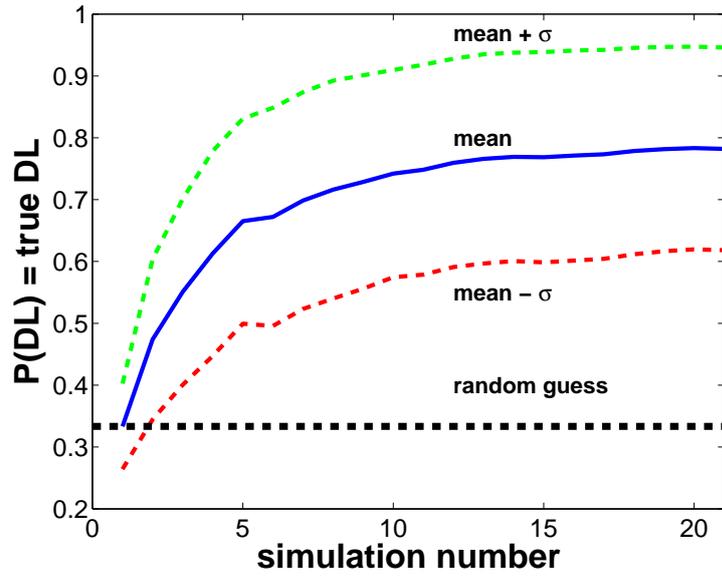
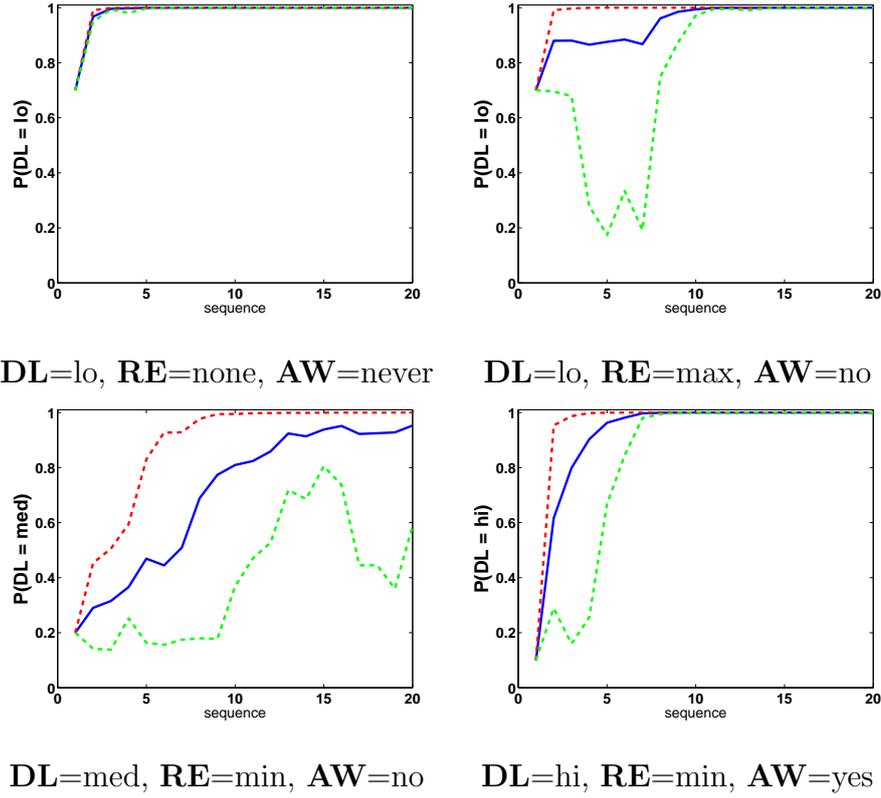
<b>DL/RE/AW</b>	PO- MDP	Heur- istic	Null	CG	CE	fo- MDP
lo/none/never	3.8±1.2	-1.1±0.9	-2.0±0.0	-75.2±3.2	6.8±0.6	9.1±0.4
lo/max/no	3.0±0.5	2.3±0.7	-0.9±0.1	-92.1±4.2	2.8±1.2	6.3±0.7
lo/med/yes	4.5±1.1	3.9±0.6	0.1±0.5	-117.8±4.0	-0.2±0.7	7.4±0.7
med/max/no	1.1±1.0	1.4±0.7	0.2±0.3	-93.6±3.9	3.4±0.8	6.0±0.8
med/min/yes	5.1±0.9	6.3±0.7	3.1±1.5	-118.4±4.3	0.9±0.9	8.1±0.6
hi/med/no	7.1±0.7	5.6±0.4	0.4±0.3	-95.6±3.9	7.2±0.7	9.3±0.6
hi/min/yes	8.3±0.7	9.8±0.6	9.7±0.9	-118.5±4.3	3.7±1.0	9.1±0.7
$\rho_0$	4.9±1.1	3.8±2.8	0.9±3.3	-97±16	4.2±2.5	8.3±1.1
$\rho_\delta$	<b>4.8±0.6</b>	<b>4.6±1.0</b>	<b>0.5±2.1</b>	<b>-105±13</b>	<b>2.9±2.4</b>	<b>7.9±0.8</b>

Table 2

Mean discounted rewards gathered over 20 simulation trials, with expectations computed over 10 experiments, for the POMDP policy, the heuristic policy (Heur.), the policy that does nothing (Null), the policy that always calls the caregiver (CG), and certainty equivalence approach (CE), and the expected value for the fully observable (MDP) version. Results are shown for a sample of different simulated user attitudes (values of **DL**, **RE**, and **AW**). The bottom two lines give the two expectations: unweighted ( $\rho_0$ ) and weighted by the initial belief ( $\rho_\delta$ ), as described in Section 4.2.

experiments for representative user types, comparing the POMDP policy, the four heuristic policies, and the MDP upper bound. Table 2 also shows the average over all user types, showing that overall, the POMDP policy performs best, but not significantly better than the heuristic or certainty equivalent (CE) policy, while the call-caregiver policy (CG) is an expensive lower bound. For particular user types, we see that the CE approach does better if the user starts the trial less aware. This is because the CE approach uses a more aggressive prompting strategy due to the collapse of the belief to a single state. For example, if the belief state is close to uniform, the optimal (POMDP) policy may be to wait and see what the user does, to try to gain some information, whereas the CE approach will commit to some state, possibly causing a prompt to be issued. The CE strategy works poorly when the user is more aware. The Null policy works best when the user is least demented and most aware (since doing nothing is close to optimal in this case).

Figure 10 shows the progression of the belief that the *dementia\_level* is equal to the true dementia level over the 20 simulations, averaged over 10 experiments. The maximum, mean and minimum values at each time step are shown. We see that for the extreme dementia/responsiveness/awareness levels, the POMDP learns the correct dementia level quickly. However, for intermediate dementia levels, the POMDP learns more slowly. This is reasonable since for these



average over all user attitudes

Fig. 10. Top 4 plots: Progression of dementia level estimates over 20 simulations for different user attitudes, with the solid line indicating the mean and the dashed lines giving the max and min over 10 experiments. Bottom plot: belief in dementia level being the true **DL** value over 20 simulations, averaged over all user attitudes, with dashed lines giving the first standard deviation, and shown relative to the  $\frac{1}{3}$  probability that would be obtained using random guessing.

intermediate cases, behaviors that could be in either extreme may be observed.

### 5.3 User Trial Results

#### 5.3.1 User Trial Results: Examples

Figure 11 shows key frames and belief states from an example sequence of subject 4 (Avg. MMSE of 11.5) during trial 34 (the first day of the first system trial). In this case, the user independently uses the soap (Figure 11-a), then turns the water on and rinses her hands (11-b). The system correctly infers she is in planstep *g* by 11-c. It gives her time to rinse, then her awareness shifts to *no*, and it prompts her to turn the water off, using a video prompt in Figure 11-c. She responds, but then gets momentarily stuck with her hands at the taps during the act of turning them off (11-d). The system gives her a further prompt to turn the taps off (11-e), cuing her to continue to the towel. She follows this prompt, using the towel to finish the task (11-f).

Figures 12 and 13 show key frames and belief states from an example sequence of subject 8 (Avg. MMSE of 10.5) during trial 59 (during the second system trial). The user immediately turns on the water (Figure 12-a) and rinses her hands, but the system has low certainty that the tap was used. The user then proceeds to dry her hands (12-b), and the system has an even distribution over plansteps *a* or *b* showing that it is uncertain about whether the water is on or off. It is, however quite certain that the user awareness is not *yes*, so prompts for soap use with minimal specificity after 31 seconds (the least costly option given the current *responsiveness* belief). The user does not respond to this prompt, and the *responsiveness* shifts towards *video* or *max*. After a timeout, the system tries the most specific (video) prompt to use soap (Figure 12-c). Given the user’s current *dementia\_level*, this is potentially a more effective prompt than a maximal prompt. Again, the user does not respond, and the system’s belief is that the user has awareness *never* and responsiveness *none*, and so calls for human assistance after about 70 seconds (Figure 12-d). The caregiver intervenes and points explicitly to the soap which gets the user to use soap (Figure 12-e), and the belief is “collapsed” to planstep *e* (by the technician behind the scenes) after 92 seconds (Figure 13-a). Notice how, in Figure 12-d, one tracker starts to track the caregiver’s right hand when one of the user’s hands is completely occluded. This is not a problem for the system, which is temporarily paused during caregiver interventions. The tracker automatically resets about a second after the occluding hand disappears.

The user then completes the trial during which she is given two prompts to turn the water on (Figure 13-b), then off again (Figure 13-d). These prompts happened because the user took a bit too long after using soap, and during

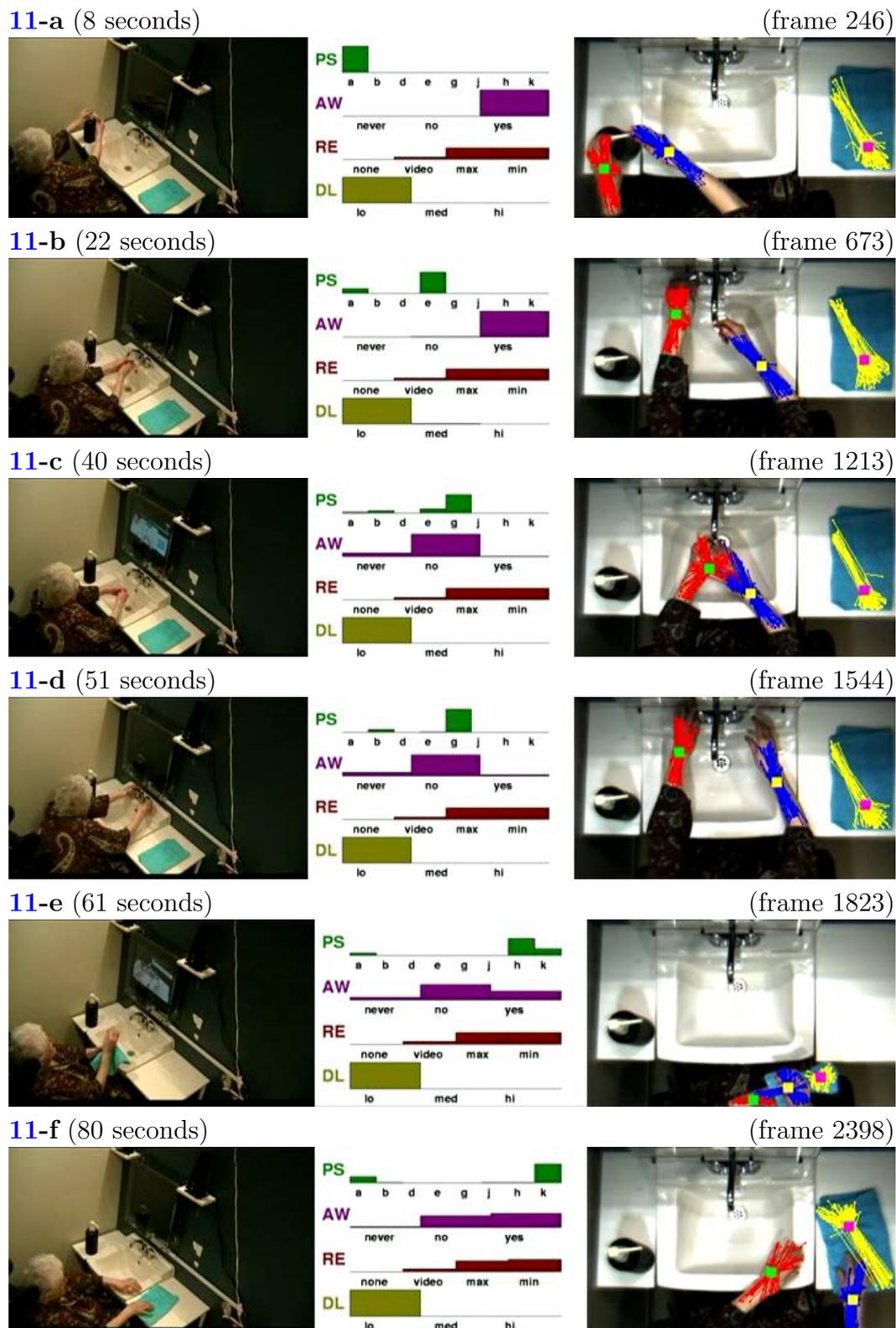


Fig. 11. Key frames from subject 4, trial 34, showing (right) the overhead video and flock trackers (center) the marginal belief state over planstep and user attitude, and (left) the camcorder view (not used by the system). See text for description.

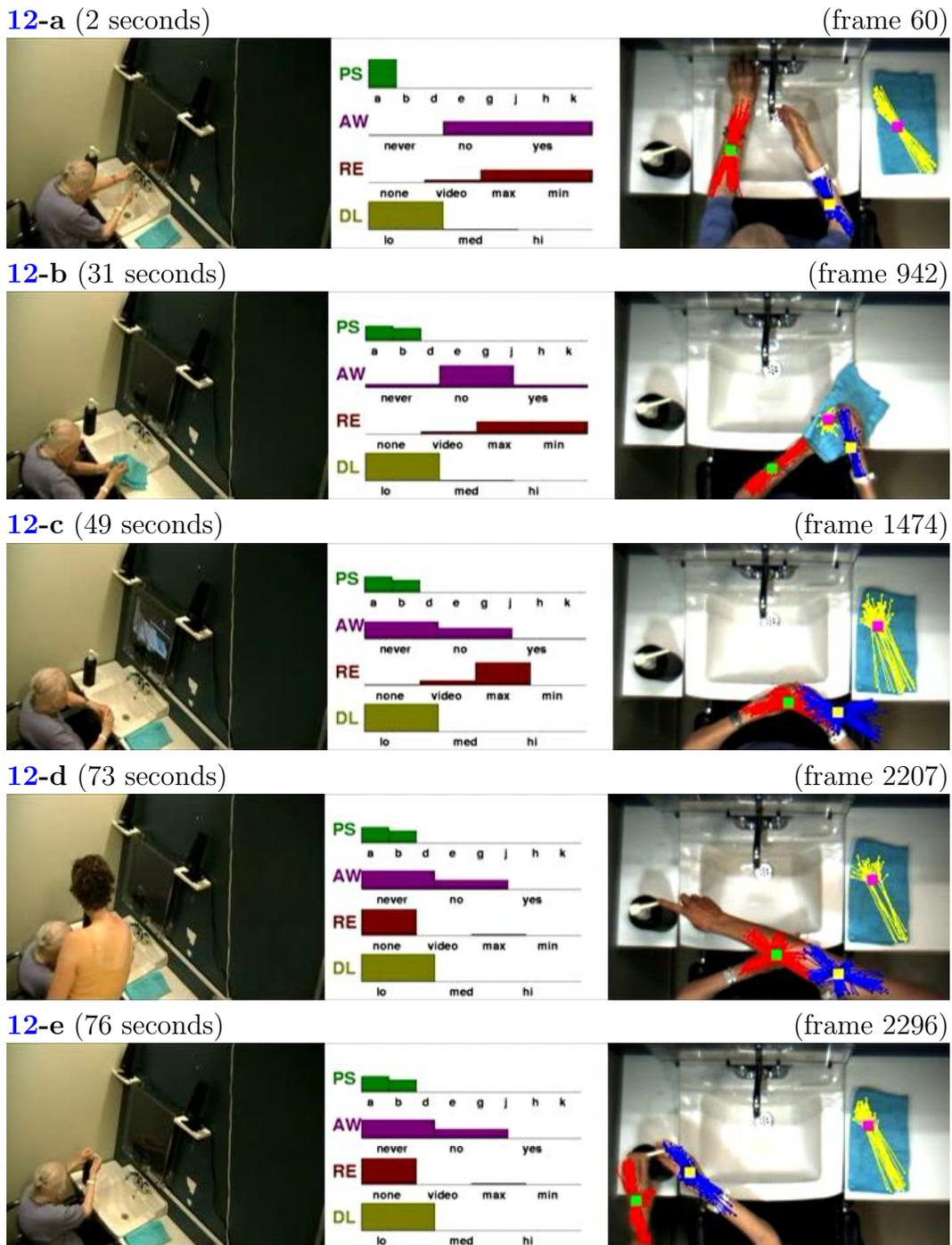
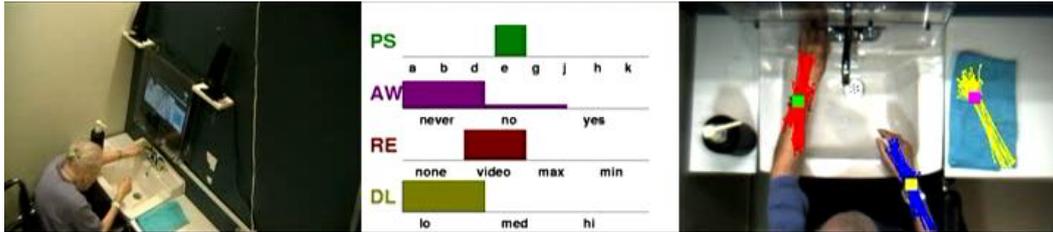


Fig. 12. Key frames from subject 8, trial 59, (part I - continued in Figure 13), showing (right) the overhead video and flock trackers (center) the marginal belief state over planstep and user attitude, and (left) the camcorder view (not used by the system). See text for description.

rinsing, causing the system to opt for a prompt. However, in this case, the prompts started simultaneously with the user beginning the behavior that

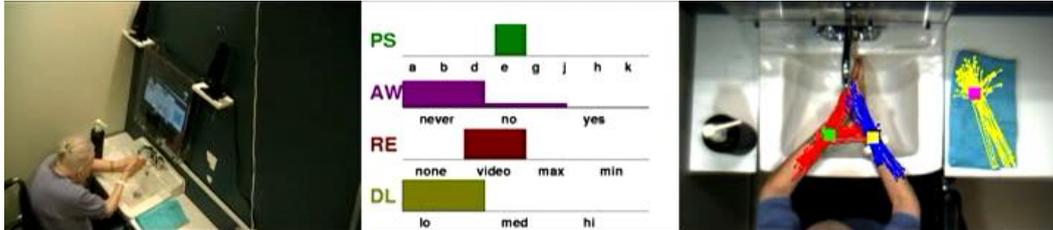
13-a (92 seconds)

(frame 2775)



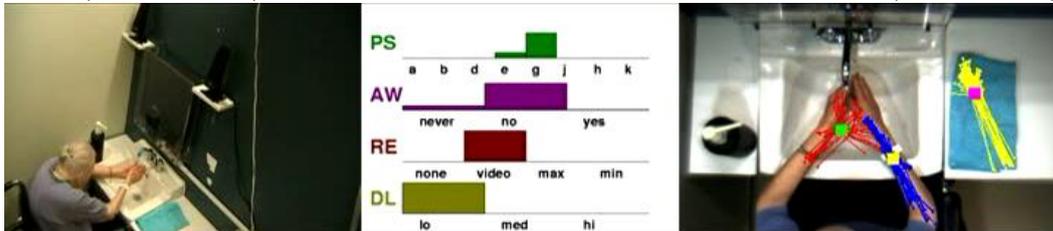
13-b (98 seconds)

(frame 2941)



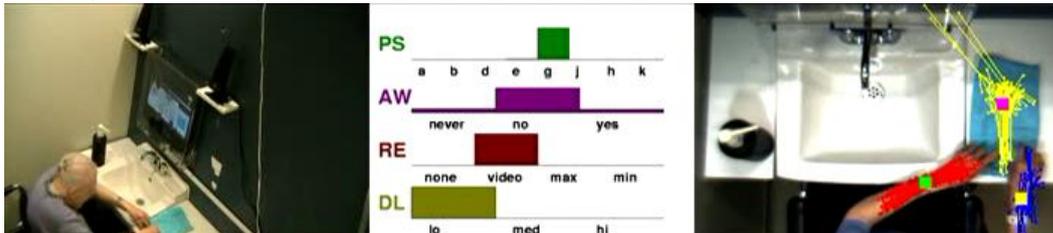
13-c (121 seconds)

(frame 3631)



13-d (127 seconds)

(frame 3808)



13-e (129 seconds)

(frame 3886)

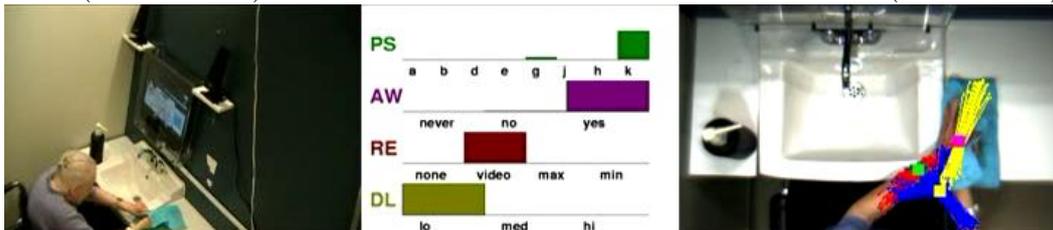


Fig. 13. Key frames from subject 8, trial 59 (part II - continued from Figure 12).

was prompted. It is unclear whether she was actually responding to prompts, or whether the prompts were played just slightly too late to make a difference. The system interprets this as the user responding to prompts, but it may be giving itself too much credit. The final frame at (Figure 13-e) shows the system has correctly estimated that the user has finished the task.

### 5.3.2 User Trial Results: Annotations

		Most likely behavior (POMDP)					
		away	sink	wet	taps	soap	dry
Annotation	N/A	33	53	11	31	23	107
	away	142	103	7	7	3	56
	sink	62	277	89	79	30	43
	wet	10	35	276	44	3	0
	taps	2	18	46	226	23	14
	soap	0	33	2	3	93	3
	dry	19	48	0	18	27	523

Table 3

Confusion matrix showing the most likely behavior given the POMDP belief (columns) correlated with the annotations (rows), averaged over all user trials. The data in row  $n$  is taken from all timesteps in which the behavior annotation was  $n$ . The darkly shaded entries are the maximum for each row.

Table 3 shows a confusion matrix for a comparison of the annotated behaviors with the POMDP’s best estimate. The entry at row  $i$ , column  $j$  is the number of times the most likely behavior given the POMDP’s belief was  $j$  when the handwashing scenario was annotated as behavior  $i$ . The largest number in each row is highlighted. We see that the behavior estimates are fairly good, with some confusion between *away* and *sink*, and the most confusing annotated behavior being *sink*. This is because the *sink* behavior can most easily be interpreted as either *wet*, *taps* or *away*: the differences in hand positions are small. The annotation of *N/A* was fairly evenly distributed across beliefs for the POMDP, most often being interpreted as *dry*.

### 5.3.3 User Trial Results: Rewards

Table 4 shows the average reward gathered during the user trials, for each of the six users. The two columns show the rewards based on the POMDP’s belief state,  $V(b)$ , and on the manual annotations,  $V(s^*)$ , as defined in Equations 10 and 11, respectively. The averages can be compared to the POMDP simulation results shown in the leftmost column of Table 2, that yielded a weighted average score of 4.8. We can see that the POMDP policy achieves nearly 80% of the value it does in simulation, implying that our POMDP model is a fairly good model of this domain, as far as its ability to be used to achieve value goes.

subject	expected	actual
	reward	reward
	$V(b)$	$V(s^*)$
1	1.84	0.51
3	7.66	11.77
4	3.93	3.22
5	1.00	1.12
6	5.22	3.41
8	2.73	2.42
avg.	3.82	3.74

Table 4

Discounted rewards gathered during actual user trials. The first column shows the expected reward over all sequences based on the POMDP’s belief state (the POMDP’s expected reward, see Equation 10). The second column shows the actual reward gathered, based on the manual annotations, averaged over all sequences (Equation 11).

subject number	MMSE Before	MMSE After	MMSE Average	final estimated expected dementia level	
				low	med
1	5	3	4	0.9992	0.0008
3	12	18	15	0.986	0.0136
4	10	13	11.5	0.9998	0.00002
5	19	20	19.5	0.016	0.984
6	12	14	13	0.9997	0.00024
8	11	10	10.5	0.9997	0.00024

Table 5

Mini-mental state examination (MMSE) scores before and after the trials, the average value of the MMSEs, and the final estimated expected dementia level (**DL**), as computed by the POMDP, using corrected dementia level belief propagation across trials. The belief is shown as a distribution over **DL**=low,med only.

### 5.3.4 User Trial Results: Estimated Dementia Level and MMSE Results

Table 5 shows the final estimated dementia level (**DL**), compared to the Mini-Mental State Examination (MMSE) scores for each subject before and after the trials. Figure 14 shows the estimated level of dementia over the course of the trials, with the average MMSE scores shown as well. First, we see from Figure 14 that the **DL** variable converges over the course of the 21

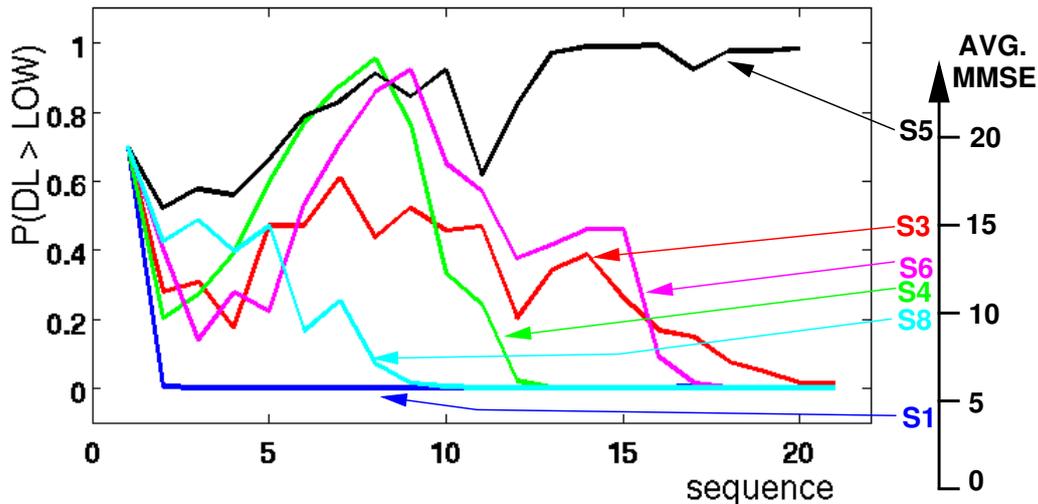


Fig. 14. Convergence of the estimated level of dementia.

handwashing sequences in the user trial. Most subject’s **DL** converged to **DL=low**, and only one (subject 5, MMSE=19.5) converged to **DL=med**. This is encouraging, as it indicates that the **DL**’s measure of functional performance in handwashing may correlate with the MMSE scores, although we cannot make any further conclusions based on this limited data. That is, this data is not inconsistent with the hypothesis that the **DL** correlates with the MMSE. It appears, however, that since most subject’s **DL** converged to **DL=low**, the model needs to be modified to include a “lower” level of **DL**. Perhaps this additional range will result in a greater spread of final converged **DL** values across this population of users, allowing for more precise correlation analyses.

## 6 Discussion and Related Work

The handwashing system described in this paper makes contributions both in computer vision modeling of human activities, and in decision making under uncertainty. However, the system is most significant for its combination of both of these contributions into an integrated, real-time, working system that has been tested on its intended user population. There are several other intelligent systems currently being developed for the older adult population. These include the Aware Home Project [26], the Assisted Cognition Project [27], Nursebot Project [28], the adaptive house [29], and House.n [30]. These projects are similar to the work described in this paper in that they incorporate AI and a decision-theoretic approach. In particular, the Autominder System [31], one aspect of the Nursebot Project, applies a POMDP in the development of the planning and scheduling aspect of the system [28]. However, these systems are mainly used as scheduling and memory aids, and do not incorporate user attitude modelling or planning for prompting. Furthermore, they have not at-

tempted a full integration of the sensing and planning components in a unified framework such as we describe here.

Computer vision research into human behavior tracking and understanding is a very active field. A notable example is a system developed to monitor hand washing quality [36], where the processes of lathering and rinsing are modeled as a detailed sequence of steps. Systems have been developed for inferring human activities [32,33,34,35], but few have used this information to interact with users. Instead, most focus on tracking and recognising human behaviors as a task in and of itself, without explicitly considering any larger goals. Exceptions are the Nursebot [28], and the adaptive house [29], but these systems have used limited computer vision.

Our tracking system is situated in a large computer vision literature on the pervasive problem of tracking an object in the presence of occlusions and distractions. Objects to be tracked usually have some consistent features, are spatially compact, and move cohesively. Typical tracking methods use some model of the appearance of an object to be tracked, and fit the model to the object over time. However, in many applications, the object's shape and appearance may change over the course of a sequence. For example, human hands change shape and velocity fairly quickly, differences which must be accounted for. Our method uses a generic type of model: a *flock* of features [6]. Flocking concepts have been applied in computer graphics [37,38], and in deterministically tracking an object with a moving camera using KLT features [6]. Our previous work described an approximate Bayesian sequential tracking method that used flocks of features to implement spatial, feature and velocity cohesiveness constraints [3]. We used a Monte Carlo approximation with a data-driven proposal distribution [8]. The idea of a multi-modal proposal was used successfully in [13], where an Adaboost process generated particles for the proposal distribution of a mixture of particle filters. The particles were *adopted* only if their distribution overlapped with one of the particle filter's mixture components. We found our tracking method to work fairly well, and is well suited for integration into the generic planning modules, as it makes few assumptions about the shapes of objects to be tracked, and so requires less human design and engineering a-priori.

Partially observable Markov decision processes (POMDPs) [39] provide a rich framework for planning under uncertainty. In particular, POMDPs can be used to robustly optimize the course of action of complex systems despite incomplete state information due to poor or noisy sensors. For instance, in mobile robotics [40], spoken-dialog systems [41] and vision-based systems [42], POMDPs can be used to optimize controllers that rely on the partial and noisy information provided by various sensors such as sonars, laser-range finders, video cameras and microphones. The handwashing system described here has led to a number of contributions to the POMDP literature. We have shown

how to solve POMDPs with continuous or large observation spaces [16], and how to approach the handwashing problem with Bayesian reinforcement learning [43]. This paper specifically introduces a number of methods for speeding up or improving the accuracy of the point-based solution methods introduced in [44,20,19].

## 7 Conclusions and Future Work

This paper described a system for assisting a person with dementia complete the task of handwashing that combines a flexible object tracker with monitoring and decision making using a partially observable Markov decision process (POMDP). The system was demonstrated during an eight-week user trial in Toronto, Canada, in summer 2007. The results showed that the system can provide assistance to this user group. However, a number of important challenges remain.

The first challenge is endowing assistive systems such as this one with the ability to **adapt** to users, both in the short and long term. Adaptivity implies the ability to adjust or modify the interaction strategy with a user over time, based on the changing abilities, needs or preferences of the user. Alzheimer’s disease, as with many forms of dementia, is progressive: the capabilities and needs of sufferers change over time, requiring assistive technologies to adapt in order to provide sustained support. Our current system only has a limited adaptivity in that it can only learn a user’s attitude based upon a fixed dynamics model. We have begun investigating Bayesian reinforcement learning methods in the context of handwashing assistants to make this system adaptive to users and able to learn their specific patterns of behaviors [43]. This learning method comes with a high complexity cost, however, and work is currently ongoing into reducing this. We believe that this complexity challenge is related to a second challenge: ensuring the computer vision models can generalise to other tasks. Adaptivity depends on input generality, since constraints posed by input specificity will reduce the potential for exploration by the system. At the same time, adaptivity is a key to reducing complexity, since new representations can be acquired “on the fly” and so the system does not need to take all possibilities into account a priori. We are currently beginning to address both of these key challenges, and will leverage their inter-dependence to develop novel algorithms for adaptivity and for computer vision.

A second major challenge is the specification or learning of the reward function. We currently specify this reward function by hand. However, the reward function encapsulates a great deal of prior information from carers and users, that should be carefully elicited from the target population. The benefit of framing the problem using a decision theoretic model is that it provides a the-

oretically well founded model within which we can start to investigate questions of preference and value tradeoffs that are inherent in cognitive assistive technologies.

A number of other avenues are open for future work.

- In this paper, we described very preliminary results showing that a hand-washing assistant could be used to monitor general user health (e.g., dementia level). We hope that this monitoring capability can be extended and improved, rendering these technologies doubly useful to carers and users.
- A benefit of the system we have developed is its ability to generalise to other ADL. We are currently looking at implementing the same system for other important washroom ADL, such as toothbrushing, and eventually, toileting.
- We are investigating methods for learning the model from data, and for integrating the tracker uncertainty into the decision making.
- Using stereo depth information for tracking
- The prompts are currently “canned” and do not change over time. We are investigating methods for making the prompts adaptive to specific users.
- The system currently has no “ears”, and so cannot respond when users say something (which they often do). We plan to add some speech recognition capabilities to the system to overcome this.
- Finally, we are looking into other non-invasive sensors to provide more inputs to the POMDP models. One example is the use of sound (microphones) to detect if the water is running or not.

**Acknowledgments.** Research supported by the American Alzheimer’s Association, the Toronto Rehabilitation Institute, and Intel Corporation. We thank Jen Boger, Dave Giesbrecht, Kate Fenton, Zhonghai Li, Rozanne Wilson. Thank you to the staff, residents, and family members at the Lakeside Long Term Care Centre (Toronto Rehabilitation Institute) for their participation and support.

## References

- [1] A. Mihailidis, G. R. Fernie, J. C. Barnebel, The use of artificial intelligence in the design of an intelligent cognitive orthosis for people with dementia, *Assistive Technology* 13 (2001) 23–39.
- [2] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, A. Mihailidis, A decision-theoretic approach to task assistance for persons with dementia, in: *Proc. IEEE International Joint Conference on Artificial Intelligence*, Edinburgh, 2005, pp. 1293–1299.
- [3] J. Hoey, Tracking using flocks of features, with application to assisted handwashing, in: M. J. Chantler, E. Trucco, R. B. Fisher (Eds.), *Proceedings*

of the British Machine Vision Conference, Vol. 1, Edinburgh, Scotland, 2006, pp. 367–376.

- [4] J. Hoey, A. von Bertoldi, P. Poupart, A. Mihailidis, Assisting persons with dementia during handwashing using a partially observable Markov decision process, in: Proceedings of the International Conference on Vision Systems, Mar., 2007, <http://bi coll.ub.uni-bielefeld.de/volltexte/2007/12>.
- [5] A. Mihailidis, J. Boger, M. Candido, J. Hoey, The coach prompting system to assist older adults with dementia through handwashing: An efficacy study, *BMC Geriatrics* 8 (28).
- [6] M. Kölsch, M. Turk, Fast 2d hand tracking with flocks of features and multi-cue integration, in: IEEE Workshop on Real-Time Vision for Human-Computer Interaction, Washington, DC, 2004.
- [7] M. Isard, A. Blake, A mixed-state condensation tracker with automatic model-switching, in: Proc 6th Int. Conf. Computer Vision, 1998.
- [8] M. Isard, A. Blake, ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework, *Lecture Notes in Computer Science* 1406 (1998) 893–908.
- [9] A. Doucet, N. de Freitas, N. Gordon (Eds.), *Sequential Monte Carlo in Practice*, Springer-Verlag, 2001.
- [10] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (11) (2005) 1805–1918.
- [11] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing* 50 (2) (2002) 174–189.
- [12] M. Isard, A. Blake, Condensation – conditional density propagation for visual tracking, *International Journal of Computer Vision* 29 (1) (1998) 5–28.
- [13] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, D. G. Lowe, A boosted particle filter: Multitarget detection and tracking, in: Springer-Verlag (Ed.), *Proceedings of European Conference on Computer Vision (ECCV)*, Vol. 3021 of LNCS, 2004, pp. 28–39.
- [14] W. S. Lovejoy, A survey of algorithmic methods for partially observed Markov decision processes, *Annals of Operations Research* 28 (1991) 47–66.
- [15] J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, A. Mihailidis, A planning system based on Markov decision processes to guide people with dementia through activities of daily living, *IEEE Transactions on Information Technology in Biomedicine* 10 (2) (2006) 323–333.
- [16] J. Hoey, P. Poupart, Solving POMDPs with continuous or large discrete observation spaces, in: Proc. International. Joint Conference on Artificial Intelligence, 2005, pp. 1332–1338.

- [17] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: Analysis and applications, *Machine Learning* 32 (1) (1998) 41–62.
- [18] M. F. Folstein, S. E. Folstein, P. R. McHugh, Mini-mental state: A practical method for grading the cognitive state of patients for the clinician, *Journal of Psychiatric Research* 12 (3) (1975) 189–198.
- [19] P. Poupart, Exploiting structure to efficiently solve large scale partially observable markov decision processes, Ph.D. thesis, University of Toronto, Toronto, Canada (2005).
- [20] M. T. J. Spaan, N. Vlassis, Perseus: Randomized point-based value iteration for POMDPs, *Journal of Artificial Intelligence Research* 24 (2005) 195–220.
- [21] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, F. Somenzi, Algebraic decision diagrams and their applications, in: *Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, Santa Clara, California, United States, 1993, pp. 188–191.
- [22] R. D. Smallwood, E. J. Sondik, The optimal control of partially observable Markov processes over a finite horizon, *Operations Research* 21 (1973) 1071–1088.
- [23] J. Hoey, R. St-Aubin, A. Hu, C. Boutilier, SPUDD: Stochastic planning using decision diagrams, in: *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, 1999, pp. 279–288.
- [24] R. St-Aubin, J. Hoey, C. Boutilier, APRICODD: Approximate policy construction using decision diagrams, in: *Neural Information Processing Systems*, Vol. 14, 2000, pp. 1089–1095.
- [25] A. Mihailidis, J. Boger, M. Candido, J. Hoey, The use of an intelligent prompting system for people with dementia, *ACM Interactions* 14 (4) (2007) 34–37.
- [26] E. Mynatt, I. Essa, W. Rogers, Increasing the opportunities for aging in place, in: *Proceedings of ACM Universal Usability Conference*, Arlington, VA., 2000.
- [27] H. Kautz, L. Arnstein, G. Borriello, O. Etzioni, D. Fox, An overview of the assisted cognition project, in: *AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, Edmonton, 2002.
- [28] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, S. Thrun, Towards robotic assistants in nursing homes: Challenges and results, *Robotics and Autonomous Systems* 42 (3–4).
- [29] M. C. Mozer, Lessons from an adaptive house, in: D. Cook, R. Das (Eds.), *Smart environments: Technologies, protocols, and applications*, J. Wiley and Sons, Hoboken, NJ, 2005, pp. 273–294.
- [30] S. S. Intille, The goal: smart people, not smart homes, in: *Proc. of the International Conf. on Smart Homes and Health Telematics*, IOS Press, 2006.

- [31] M. E. Pollack, Planning technology for intelligent cognitive orthotics, in: AIPS, Toulouse, 2002, pp. 322–331.
- [32] C. R. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: Real-time tracking of the human body, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (7) (1997) 780–785.
- [33] N. Oliver, A. Garg, E. Horvitz, Layered representations for learning and inferring office activity from multiple sensory channels, *International Journal on Computer Vision and Image Understanding* 96 (2004) 163–180.
- [34] L. Liao, D. Fox, H. Kautz, Learning and inferring transportation routines, in: Proc Nineteenth National Conference on Artificial Intelligence (AAAI '04), San Jose, CA, 2004, pp. 348–353.
- [35] N. T. Nguyen, D. Q. Phung, S. Venkatesh, H. H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model, in: Proc. CVPR, San Diego, CA, 2005.
- [36] D. F. Llorca, F. Vilarino, J. Zhou, G. Lacey, A multi-class svm classifier for automatic hand washing quality assessment, in: Proc. of the British Machine Vision Conference, 2007.
- [37] W. T. Reeves, Particle systems - a technique for modeling a class of fuzzy objects, *ACM Transactions on Graphics* 2 (2) (1983) 91–108.
- [38] C. W. Reynolds, Flocks, herds, and schools: A distributed behavioral model, in: *Computer Graphics*, Vol. 21 (4), ACM SIGGRAPH, 1987, pp. 25–34.
- [39] K. J. Åström, Optimal control of Markov decision processes with incomplete state estimation, *Journal of Mathematical Analysis and Applications* 10 (1965) 174–205.
- [40] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, V. Verma, Experiences with a mobile robotic guide for the elderly, in: AAAI, Edmonton, 2002, pp. 587–592.
- [41] J. Williams, P. Poupart, S. Young, Factored partially observable Markov decision processes for dialogue management, in: Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Edinburgh, Scotland, 2005, pp. 76–82.
- [42] T. Darrell, A. P. Pentland, Active gesture recognition using partially observable Markov decision processes, in: 13th IEEE Intl. Conference on Pattern Recognition, Vienna, Austria, 1996.
- [43] P. Poupart, N. Vlassis, J. Hoey, K. Regan, An analytic solution to discrete Bayesian reinforcement learning, in: Proceedings of the 23rd International Conference on Machine Learning (ICML), Pittsburgh, Pennsylvania, 2006.
- [44] J. Pineau, G. Gordon, S. Thrun, Point-based value iteration: an anytime algorithm for POMDPs, in: IJCAI, Acapulco, Mexico, 2003.