Exploiting Time Series Data for Task Prediction and Diagnosis in an Intelligent Guidance System

Hayley Borck, Steven Johnston, Mary Southern, and Mark Boddy

Adventium Labs, 111 Third Ave South Suite 100, Minneapolis, MN 55401

Abstract. Time series data has been exploited for use with Case Based Reasoning (CBR) in many applications. We present a novel application of CBR that combines intelligent tutoring using Augmented Reality (AR) and prediction. The MonitAR system, presented in this paper, is intended for use as an intelligent guidance system for astronauts conducting complex procedures during periods of a communication time delay or blackout from Earth. Our approach takes advantage of the relational nature of time-series data to detect a *task* that the user is completing and diagnose the issue when the user is about to make a mistake.

1 Introduction

Astronauts are trained in a myriad of different procedures ranging from maintenance to emergency medicine. To alleviate mistakes during stressful situations, guidance during such procedures is advantageous. However with longer spaceflight missions comes delays or blackouts in communication with Earth. Such instances would benefit from a training and guidance system that is able to direct the user during the procedure and guide them away from potential mistakes before one is committed. For the duration of this paper we refer to a *procedure* as a NASA procedure for complex tasks and a *plan* as the representation of a procedure in a planning language. A *task* is the smallest step in a plan. Further, a case in MonitAR is comprised of a *problem*, represented by a series of time step features, and a *solution* that is the previously mentioned task.

MonitAR is a training and guidance system that predicts the task the user's currently completing. If MonitAR predicts the user will make a mistake, it guides the user back to the correct task using visual cues. The system monitors a user's activity while completing a task taking data at set intervals. The data is collected through the camera of an Augmented Reality (AR) smart glasses device. Features are created using the positions of objects in relation to other objects within the view of the AR device. Over time the relationship between the objects indicate the task the user is completing. MonitAR uses partial data to predict the task the user is completing so as to eliminate potential mistakes. A diagnosis of how the user is completing the task incorrectly, such as if the user is completing tasks in the wrong order, is done to create visual cues. The sequential nature of time series data is manipulated during diagnosis which aids in determining the mistake. Learning is employed when the user indicates to

In Proceedings of the ICCBR 2016 Workshops. Atlanta, Georgia, United States of America

the system that they are completing the task in a previously unknown (to the system) way. The MonitAR system aims to aid astronauts while completing procedures in which the astronaut is not an expert or when the astronaut is performing under stress and would normally be given precise instruction via experts on Earth. This can be generalized to aid in any situation where the user is not an expert in the procedure.

The remaining sections of the paper are broken down as follows. Section 2 discusses related work, section 3 gives an overview of the MonitAR system architecture, section 4 describes how time series data is represented in our system. In section 5 the prediction component of the system is described, section 6 details the diagnosis component of the system, and finally the experiment and conclusion are discussed in sections 7 and 8.

2 Related Work

Prediction and recognition of users and opponents has been a well researched area in recent years. Less so, however, is the prediction of the task the user is completing. The intelligent tutoring community has shown great strides in modeling the user, and determining how best to help them through a task. The AR community has been doing guided procedures for some time in numerous domains. We believe our system which combines intelligent tutoring and prediction using AR is the first of its kind. Given the current research and state of technology this area of research seems likely to flourish in the coming years.

2.1 Prediction and Recognition

Prediction and recognition of human activity using visual data is an active area of research. Our approach to prediction leans on this existing body work. Pei et al. [8] and Auslander et al. [4] in particular have created recognition systems from visual data. Our problem is made easier than the usual plan or intent recognition domains because in this domain we know which task the user *should* be completing. Prediction coupled with diagnosis using CBR in the low to no communication space domain using AR is our new contribution.

Synnaeve et al. [9] presented a bayesian programming approach to predict an opponent's opening strategy in RTS games. We show in our experiment that a CBR approach to predicting the current task is better than a straightforward Bayesian approach in our domain. The most similar prediction work to our own came from White et al. [14]. They describe a Capability Aware, Maintaining Plans approach in addition to a Belief, Desires, and Intentions (CAMP-BDI) system that preempts anticipated failure. Their work, however focuses on failure due to outside issues, rather than issues relating to the user's own confusion, stress, or misunderstanding of the current task. Antwarg et al. [3] showed that adding a user profiling component to an intent prediction system increases the accuracy of the prediction. We believe applying a user profiling system will aid in our system as well and intend to implement it in future work.

2.2 Training and Tutoring

The Intelligent Tutoring community has published some great work on guiding users towards better learning. Early systems such as presented by Anderson et al. [2] have shown the usefulness of AI in training. The MonitAR system does not fall into a prescribed definition of an ITS because we do not provide tutoring or training services rather we guide the user when a mistake is made. The guidance our system provides however, does use similar principles as an ITS system.

The visual cues MonitAR presents the user in order to guide them back to the correct task has similar qualities to a traditional constraint based modeling ITS, as defined by Ma et al. [7]. The visual cues we provide qualify as 'a feedback message that, when the solution state fails the satisfaction condition, advises the student of the error...'. Additionally in their survey Ma et al. [7] found that ITS systems were associated with positive effects across a wide range of domains from humanities to the sciences indicating the potential of the MonitAR system in a wide range of procedures and domains.

Grasser et al. [6] created the AutoTutor system, that helps college students learn computer literacy through a conversational tutor. This shows us that ITS systems may be helpful to users with a high level of education. Aleven et al. [1] suggests users are reluctant to seek help and that users who are at a medium level of mastery are benefited by hints given without the user asking. Admittedly our target audience, astronauts, are at a higher level of education and mastery than ITS' are generally geared for. We still believe an intelligent guidance system such as MonitAR will be beneficial and plan to complete user studies in the future that will corroborate this hypothesis.

2.3 Augmented Reality

A survey by [5] describes the current state of the art (as of 2015) in first person activity recognition through video, paying special attention to AR and wearable devices. They describe two approaches to activity recognition through AR devices as object based and motion based, which our system combines to both predict and diagnose errors. Additionally they highlight that none of the approaches are able to work in a *closed-loop* fashion by continuously learning from users, which we attempt to address. Others have used AR devices for training and guidance. Wacker et al. [11] presented an AR guidance system for image guided needle biopsy. Similarly Vosburgh et al. [10] use AR for guidance during laparoscopic surgery using CT or MRI images. AR guidance for maintenance and assembly tasks has been done by Webel et al. [12]. The MonitAR system aims to generalize to many different types of procedures encompassing the previously mentioned domains. The Westerfield et el. [13] system incorporates the intelligent tutoring techniques with AR similar to MonitAR our system however, goes one step farther in predicting mistakes and alerting the user.

3 System Overview

The full MonitAR system can be seen in fig. 1. Procedures are taken from the International Space Station (ISS). While the user executes a task in a plan the AR Interface component collects features from the camera using an object recognition library. At each time step, features are passed to the CBR Task Prediction component. During training this component collects features until the task is complete then writes the case to file. During execution a partial case is compiled and retrieval is executed at each time step. If a case is found during retrieval which is over a prediction similarity threshold the partial case, predicted case, and a case determined to represent the current task (the 'correct case') are given to the Diagnoser component. The Diagnoser merges the partial and predicted case and calculates the difference between this merged case and the correct case using *delta cases* which are discussed in later sections. This difference is used to create visual cues within the AR Interface.

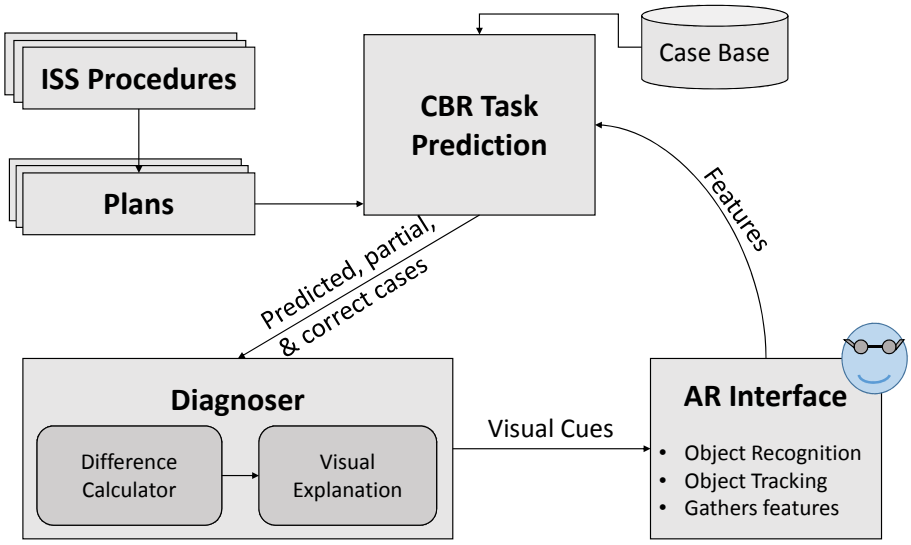

Fig. 1. Architecture of the MonitAR system

4 Representation of Time Series Data

Time series data is represented in the MonitAR system in two ways. During prediction of a task the data is represented as distance relationships between recognized objects in view. At each time step the distance of each object in view related to each other object in view averaged over the time step length as

distance features. A *time step feature* is comprised of multiple *distance features*. A short time step length of 500ms was chosen in order to collect enough data to quickly and correctly predict during relatively short tasks. The position of the hand at the beginning and end of the time step is also annotated and added to the *time step feature*. The annotation of hand position enables the system to reason on how the hand is moving within the length of the time step. A sample case using partial information that indicates the hand reaching towards obj_1 and away from obj_2 in two time steps can be seen in Fig.2.

TimeStepFeature @ time t *HandPosBeg:* [0,0,0] *HandPosEnd:* [25,25,0] *DistanceFeature* Hand:Obj$_1$ Dist: 100 *DistanceFeature* Hand:Obj$_2$ Dist: 50	*TimeStepFeature* @ time t$_1$ *HandPosBeg:* [25,25,0] *HandPosEnd:* [50,50,0] *DistanceFeature* Hand:Obj$_1$ Dist: 75 *DistanceFeature* Hand:Obj$_2$ Dist: 75

Fig. 2. A sample partial case showing two time step features

During diagnosis of the predicted mistake a *delta case* is created by merging adjacent time step features. To merge time step features the distance in each distance feature of a time step feature for time *t+1* is subtracted from the matching (containing the same objects) distance feature from the time step feature at time *t*. Using the case in 2 as an example, the merged time step feature for the delta case would have two distance features. The distance feature containing the hand and obj_1 would have a distance of -25. The distance feature containing the hand and obj_2 would have a distance of 25. Delta cases represent the movement of recognized objects between time steps and provide a way of determining the relationship between objects over time. See section 6 for more detail on delta cases and the diagnosis process.

To handle faulty sensors we employ filters using heuristics based on the way the physical world works. When an object which was previously recognized is not recognized in the current time step, distance features are added to the time step feature at the same position as previously seen. In some instances, for example when a hand grabs an object and occludes it, this heuristic fails. To combat this when a missing object is recognized in a different location than it was previously and near an object which can move it, such as a hand, distance features are added to each time step feature where that object was missing using the same distance relations as the object that presumably moved it. Lastly, the user can introduce camera jitter due to slight movements even when standing 'still'. We ran a short experiment and found that a typical user will sway up to 15mm so

we accounted for this possible distance change in the similarity function. These input filters solve the most significant issues found with the camera, occlusion, and the object recognition library.

5 Prediction

During execution of a task, time step features are created by the AR system and handed to the CBR Task Prediction component. After a time step feature has gone through the input filters, a set of n cases are retrieved from the case base using the similarity function. The similarity function is comprised of two parts. The first part is a weighted sum of distances between objects. The second part consists of the distance from the current and projected hand position of the partial case q to the current and next hand position of the case base case c. Sequentiality of the time steps enable a projection of hand positions which give the system more information for the similarity function to use, allowing a quicker prediction. These two parts are weighted and added together to create the similarity score. The full equation is shown in Eqn-1. For the weighted sum of distances we choose to weight time step features using linearly ascending weight, γ, to model that the later time steps better indicate what the user is trying to accomplish. In the following equation m is the number of time step features in the case base case c, n is number of matching distance features between c and q, Qd_f and Cd_f indicate the distance feature in case q and c, and finally ch_f and nh_f are the current and next hand positions.

$$sim(q,c) = \alpha \frac{\sum(\gamma \frac{\sum(1-(Qd_f - Cd_f))}{n})}{m} + \beta(\zeta(ch_f) + \eta(nh_f)) \quad (1)$$

The top l cases with a similarity over a threshold t_{sim} are brought back from the case base. If any of the l cases have a greater similarity then a prediction threshold t_{pred} the top case is handed to the Diagnoser to component as a predicted case.

6 Diagnosis

The Diagnoser component is responsible for determining the difference between the predicted task and the task the user should be completing. Visual cues are created during diagnosis that show the user the deviation from the correct task via the AR Interface. The Diagnoser conducts the reuse phase of the CBR work flow to adapt the predicted case to the current situation. To do this we first merge the predicted case and the partial case to create a complete case by taking the time steps t - t_n of the partial case and adding the remaining time steps from the predicted case t_{n+1} - t_m. The cases are merged in order to give the Diagnoser the most grounded information possible, rather than relying solely on the predicted case to be similar to reality. *Delta cases* are created from this merged predicted case and the correct case for the current task.

From the delta cases we are able determine the target object of each case, by which we mean the object that the hand is moving towards. Two assumptions are made here first that there is exactly one hand represented in the case and secondly whatever the hand is doing is imperative to the task. In future iterations of this system we will generalize this to a generic type of object. To determine the target object the sum of the distance features between the hand, and each other object is found. The largest sum represents the object that the hand traveled to the fastest and therefore is the target of that delta case. Visual cues are created by finding the target objects of the correct case and the merged predicted case. In the instance that the target objects are different, a visual cue of a highlighted green box is drawn over the target object of the correct case, while a red box is drawn over the target object of the merged predicted case (fig. 3).

The difference routine reuses the time step portion of the similarity function with the delta cases. Since the delta case represents the movement of objects between time steps, the similarity function here indicates the similarity of that movement. This is opposed to when the similarity function is used in prediction where the calculation between the partial and case base cases represent the similarity between the location of static objects. The time steps that have a low similarity (or high difference) under a difference threshold are compiled to create the visual cues, the largest difference is used to create the final visual cue.

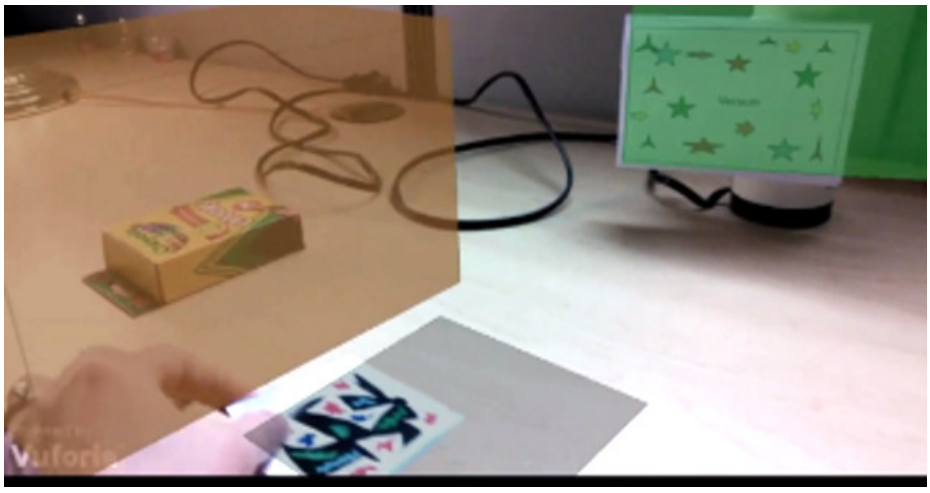

Fig. 3. MonitAR indicated the vacuum (top right) as the correct target object within the task and the crayon box (middle left) as the incorrect target object by highlighting the objects in green and red respectively. The hand (middle bottom) is highlighted in gray to indicate it is a recognized object.

We have created a series of distinct image targets (2D images) to label objects such as the hand or vacuum shown in Fig.3 to make the object recognition

task easier. The focus of this project is not intended to include development of improved object recognition algorithms.

7 Experiment

The results of our prototype MonitAR system are encouraging. For the experiment we tested whether the MonitAR system could correctly predict the task the user was completing. Our initial tests were compared against a naive Bayes task prediction. The CBR system was trained on four plans containing a cumulative thirteen tasks. One hundred cases were created during training which encompassed two users (with different handedness) completing the plans. The experiment was run using k-fold cross validation with $k = 10$.

The naive Bayes prediction is calculated during the retrieval phase replacing Eqn-1 with the following Eqn-2. In Eqn-2 c_τ is a case with task τ, and q represents the current partial case. The conditional $p(q|c_\tau)$ is the probability a case c encompasses the partial case q and has a solution of task τ. $p(q)$ is the probability that the case encompasses the partial case q. Finally $p(c_\tau)$ is the probability a case c has a solution of task τ. The top ten cases are returned during the retrieval step. The case with the highest probability, if it is over the prediction threshold t_{pred}, is the prediction. Both methods used the same prediction threshold t_{pred}.

$$p(c_\tau \mid q) = \frac{p(q|c_\tau) \cdot p(c_\tau)}{p(q)} \tag{2}$$

There was a significant difference in the percentage of correct predictions for MonitAR using the similarity Eqn-1 and the Bayesian Eqn-2 ($p < 0.0001$ using a paired t-test). MonitAR gave on average 148 more predictions than its Bayesian counterpart with an average percent correct prediction of 81% when the $t_{pred} = .6$ and $t_{sim} = .4$. Even though the propensity to report false positives is higher using MonitAR due to the sheer amount of predictions made, the gains over Bayesian retrieval, that had an average percent correct prediction of just 43% are significant. The average earliest correct prediction was better using Bayesian retrieval: 1.04 seconds vs 1.2 seconds for Bayesian and MonitAR respectively. The experiment was rerun with a $t_{pred} = .8$ and $t_{sim} = .4$ the results also showed MonitAR correctly predicting the task at a significantly higher percentage. Future experiments will be run to determine the best values for t_{pred} and t_{sim}.

If we look deeper into the results, we can see that certain tasks were easier to predict than others (Fig. 4). In particular T3 did very poorly, this can be explained by the nature of the task which asks the user to remove a battery from a power tool. To do so means the user's hand is reaching toward both the battery and the power tool for most of the case. Instances such as this will be addressed in the future with the addition of more fine grained features. Task T2 also did poorly using either method which we believe is due to the length of the task which was very short. We surmise using the Bayesian probability as a

confidence score in conjuncture with Eqn1 will bring the overall correctness and timeliness of the prediction up. This will be explored in future work.

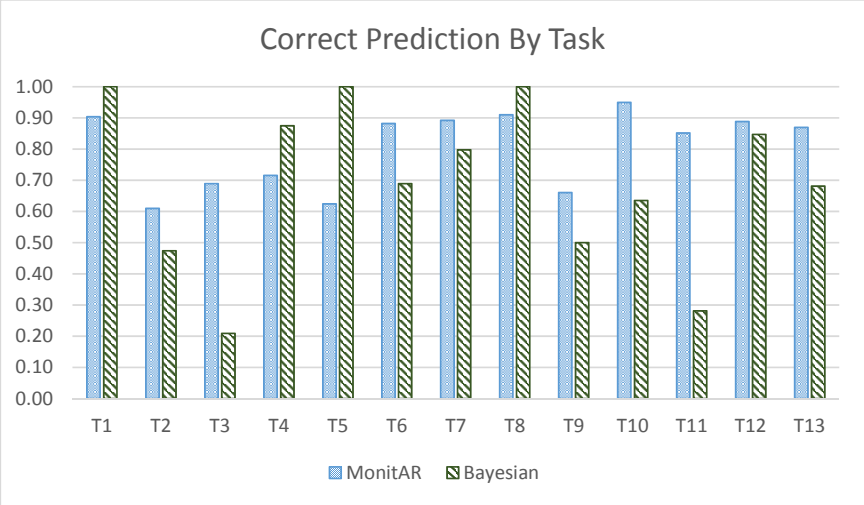

Fig. 4. Percent correct prediction by task for MonitAR and Bayesian retreival using $t_{pred} = .6$ and $t_{sim} = .4$

8 Conclusion

This paper presented early work done on the MonitAR system for task prediction and mistake diagnosis using visual cues. The system is a novel application of CBR to monitor a user's activity and give visual feedback upon the prediction of deviation to a plan. Our system leans on previous work in plan prediction and recognition and has wide applications within training and procedure guidance domains. The MonitAR system has shown promising results in prediction time and correctness when compared to other approaches. Future work will work encompass a full experimental study to determine the best thresholds to employ and weights as well as the addition to more fine grained features.

9 Acknowledgments

The material is based upon work supported by the National Aeronautics and Space Administration under Contract Number NNX16CJ22P. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

1. Vincent Aleven, Ido Roll, Bruce M McLaren, and Kenneth R Koedinger. Help helps, but only so much: research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26(1):205–223, 2016.
2. John R Anderson, C Franklin Boyle, and Brian J Reiser. Intelligent tutoring systems. *Science(Washington)*, 228(4698):456–462, 1985.
3. Liat Antwarg, Lior Rokach, and Bracha Shapira. Attribute-driven hidden markov model trees for intention prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1103–1119, 2012.
4. Bryan Auslander, Kalyan Moy Gupta, and David W Aha. Maritime threat detection using plan recognition. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pages 249–254. IEEE, 2012.
5. Alejandro Betancourt, Pietro Morerio, Carlo S Regazzoni, and Matthias Rauterberg. The evolution of first person vision methods: A survey. 2015.
6. Arthur C Graesser, Kurt VanLehn, Carolyn P Rosé, Pamela W Jordan, and Derek Harter. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39, 2001.
7. Wenting Ma, Olusola O Adesope, John C Nesbit, and Qing Liu. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, 106(4):901, 2014.
8. Mingtao Pei, Zhangzhang Si, Benjamin Z Yao, and Song-Chun Zhu. Learning and parsing video events with goal and intent prediction. *Computer Vision and Image Understanding*, 117(10):1369–1383, 2013.
9. Gabriel Synnaeve and Pierre Bessiere. A bayesian model for opening prediction in rts games with application to starcraft. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 281–288. IEEE, 2011.
10. Kirby G Vosburgh and R San Jose Estepar. Natural orifice transluminal endoscopic surgery (notes): an opportunity for augmented reality guidance. *Studies in health technology and informatics*, 125:485, 2006.
11. Frank K Wacker, Sebastian Vogt, Ali Khamene, John A Jesberger, Sherif G Nour, Daniel R Elgort, Frank Sauer, Jeffrey L Duerk, and Jonathan S Lewin. An augmented reality system for mr image–guided needle biopsy: Initial results in a swine model 1. *Radiology*, 238(2):497–504, 2006.
12. Sabine Webel, Uli Bockholt, Timo Engelke, Nirit Gavish, Manuel Olbrich, and Carsten Preusche. An augmented reality training platform for assembly and maintenance skills. *Robotics and autonomous systems*, 61(4):398–403, 2013.
13. Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. Intelligent augmented reality training for assembly tasks. In *Artificial Intelligence in Education*, pages 542–551. Springer, 2013.
14. Alan White, Austin Tate, and Michael Rovatsos. Camp-bdi: A pre-emptive approach for plan execution robustness in multiagent systems. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 65–84. Springer, 2015.