Dynamical Models for Instruction Completion and Error Recognition for NASA Physical Procedures

Steven Johnson
Department of Computer Sciences
University of Wisconsin–Madison
sjj@cs.wisc.edu

Ronak Mehta
Department of Computer Sciences
University of Wisconsin-Madison
ronakrm@cs.wisc.edu

John Cabaj
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
cabaj@wisc.edu

Abstract

TODO::: Describe the goal of the project, the data/device as well as which techniques you plan to use.

1. Introduction

Procedures are the accepted means to operate a spacecraft system or systems to perform specific functions, and consequently are at the heart of all NASA human spaceflight operations [4]. A procedure is a detailed set of instructions specifying how a piece of equipment is operated or a task is performed [3]. They are often written to be very general and to cover numerous contingencies. Procedures to operate a class of equipment (e.g., smoke detector) will differ based on make, while procedures to operate a piece of equipment will have conditional or optional steps based on configuration. As an additional complication, constraints of some procedures may be highly conditional, discretionary, or unordered. At the same time, there may be external constraints that limit how a procedure must be executed, and these constraints are not made explicit. The outcomes of NASA missions rely on crew members properly executing a multitude of these complex procedures, making procedure execution support and monitoring a critical factor that can determine success or failure measured both in terms of monetary costs as well as preventing loss of life.

There is a body of prior NASA work focused on monitoring the progress of procedures that are not physical. For instance, when instructions to systems of the ISS are sent from ground, the application ThinLayer highlights commands as they are executed to show procedure progress [3]. IPV itself also allows for manually tracking procedure progress for a crew person onboard ISS. However, to date there is little work from NASA in the realm of tracking execution status of physical procedures where crew members are manually manipulating physical objects, such as during maintenance tasks. Our goal with this project is to develop a method to computationally model a procedure to enable tracking of the execution of its steps and detection of crew errors during execution.

The inputs to our system will be a set of videos of users correctly executing one procedure (an exercise equipment maintenance task) recorded from a head-mounted egocentric camera. Using this set of videos, we will develop a technique for learning a dynamical model of the procedure that extends current methods by incorporating domain knowledge from the provided procedure documentation. We will then evaluate the model on a set of videos which are both correct and contain errors to determine the accuracy of error detection and overall instruction segmentation.

2. Related Work

Significant work has been done in the field of human action and activity recognition. In [8], Turaga presents a comprehensive overview of this work in detail. In most work, activity recognition is identified as the sum of *actions* performed in

a temporal ordering. Parametric models, particularly Hidden Markov Models, have been used with success in many applications. [9] employ them to identify whole-body tennis swings. Using background subtraction, they are able to identify the actor in the scene, and learn a model based on how the actor alone is moving over time.

Closer to our domain, [7] use two real-time HMMs to identify American Sign Language using an ego-centric camera with high accuracy.

In order to properly specify the given task, it is necessary to maintain several levels of abstraction to gain an overall representation of the procedure. It is intuitive to use three levels: task, activity, and actions where the task is the overall goal of the procedure, activity is the sub-activities making up the task, and actions are the lowest level for each activity. To this end, Petri Nets[6] are an obvious choice, given the parallel drawn between actions on an object, and transition actions which change the state of the object.

In order to provide action transitions, a proper representation of the action must be attained. Feature extraction must be performed to determine object representation at a low-level. [2] describe a method in which to identify objects as well as some coarse actions which aid in object identification. Given the object representation, we can use a Hidden Markov Model (HMM) to determine the likely action based on the objects in frame and the current action state. This provides a spatio-temporal model in which to classify the low-level actions making up the activity level Petri Nets.

The entire representation of a task as a whole is broken up into two overall levels: a Petri Net level to provide a framework for probable next actions, and a Hidden Markov Model with underlying feature representation to define low-level activities. We combine both levels to address the shortcomings of each. Petri Nets tend to only allow for more abstract actions and are difficult to train, while HMMs are more limited to recognition of a specific action and can't maintain an overall sense of the state of the task as a whole.

3. Problem Description and Techniques You Want to Use

While our project falls in the general space of activity recognition, our specific task includes key domain knowledge that we plan to leverage. Most of the work mentioned above describes tasks in which a single body is in motion against a known background with a fixed camera. Here, an ego-centric camera is used, and tasks involve the manipulation of objects within the scene. Not only is it necessary to be able to distinguish the action currently taken place, but objects in the space must be tracked as they are being manipulated. These objects may be occluded, combined, or separated throughout the task. A key part of our project will be creating a strong feature representation to extract from the raw video. [7] provide us with a strong starting point, as they use prior information about relevant objects in the scene that allow them to quickly extract these objects from irrelevant background.

Once a feature representation has been extracted, a model must be created that will allow us to learn the task in question. There has been some previous work on temporal modeling of time-variant tasks as mentioned above, but with mediocre success. Again, we will use [7] as our starting point, given the similar underlying task. Here is where we attempt to incorporate our domain knowledge, in the hope that we may be able to create a stronger model.

Our higher level activity recognition may require a significantly different model than that used for the lower-level action recognition. Because each "activity" may be correct or incorrect, and may occur in different temporal orderings, a model that incorporates logic will be necessary. Petri-Nets have been used by [1], because of their natural representation of sequencing, parallelism, and synchronization. All of these will prove to be valuable in our domain, and may allow us to better model the nature of step-by-step assembly.

4. Experimental Evaluation

Describe the data you plan to use, the baselines you want to compare (if applicable), and the evaluation metric.

If you plan to use some special equipment, mention it here. We may be able to help with servers, extra cameras, etc.

5. Conclusion and Discussion

Breakdown–what will each team-member do? Time-line of your project? Ideally, everyone should do something imaging/vision related (it's not good for one team member to focus purely on user-interface, for instance).

You full proposal should be two pages excluding references using this template.

References

[1] C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *ECCV Workshop on Conceptual Descriptions from Images*, pages 13–27, 1996. 2

[2] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011. 2

[3] J. Frank. When plans are executed by mice and men. In *Aerospace Conference, 2010 IEEE*, pages 1–14. IEEE, 2010. 1

[4] D. Kortenkamp, R. P. Bonasso, D. Schreckenghost, K. M. Dalal, V. Verma, and L. Wang. A procedure representation language for human spaceflight operations. In *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-08)*, 2008. 1

[5] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[6] C. A. Petri. Communication with automata. 1966. 2

[7] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, 1998. 2

[8] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008. 1

[9] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992. 2