Task Automation by Interpreting User Intent

PhD Proposal

Kevin R. Dixon

Thesis Committee:
Pradeep K. Khosla, Bruce H. Krogh, Maja J. Matarić, Christiaan J.J. Paredis, Sebastian B. Thrun

24 April, 2001
Introduction and Motivation

• Current development systems require traditional-programming methods.
  • Requires computer-programming and domain expertise.
Introduction and Motivation

- Most domain experts have little procedural-programming knowledge.
- Acquiring programming expertise is expensive.
- Users tend not to automate daily tasks.
- Many industrial tasks are still performed manually.
**Introduction and Motivation**

- Develop an improved paradigm for automating tasks.

- Increase user productivity by creating a more natural programming process.

- Remove computer programmers from the design loop.
Approach

• Allow users to “program” automation tasks by demonstration.

• Called “Learning by Observation” or “Programming by Demonstration”
Goals of Research

• Address uncertainty in sensing.
• Interpret intent of user demonstrations.
• Insensitive to changes in the environment.
• Incorporate multiple demonstrations to improve performance.
• Integrate the user into the design loop.
Presentation Overview

- Learning by observation overview.
- Related work.
- System design.
- Sample implementation.
- Proposed work.
- Evaluation of research.
- Expected contributions.
Learning by Observation: Inherent Problems

- Must contend with usual sensor-based uncertainty.

- Environment is dynamic.

- Many demonstrations may be required to achieve generality.
  - But available data will be sparse.
Learning by Observation: Interpreting User Intent

• In many tasks, repeating the actions of the user verbatim is not desirable.

• Humans tend to be imprecise and make unintended actions.

• The LBO system must interpret the *intent* of the user.
Learning by Observation: Definitions

- **Task**: sequence of actions designed to achieve an overall goal.

- **Subgoals**: set of states sufficient to complete the task.

- **Environment Description**: information conveying anything that could affect the task.
Learning by Observation: Related Work

- Most previous systems segment fit observations to predefined symbols called *primitives*:
  - **HMMs**: Yang, Xu, and Chen [1994]
  - **TDNNs**: Friedrich *et al.* [1996]
  - **DTW**: Ikeuchi *et al.* [1994], Matarić [2000]
  - **Ad Hoc**: Bentivegna and Atkeson [1999]

- Primitives are used to reconstruct the demonstration.

- Most do not incorporate multiple demonstrations.
Learning by Observation: Related Work

• Manually associating subgoals in the environment: Morrow and Khosla [1995].
  • Assembly tasks.

• Allowing user to modify LBO output: Friedrich et al. [1996].
  • Editing predicate-calculus statements.
Problem Description

- **Input**: Observations from repeated user task demonstrations and environment information.
- **Output**: Generative model (production program, controller, etc.)
System Design

- System is broken up into two main phases.
- Data collection.
- Analysis, synthesis, and production.
System Design: Determine Environment Configuration

- Description of all objects that could affect the task.
System Design: Observe User

- We plan to use cameras to observe users performing tasks.
- Other modal inputs may be considered.
  - We want to instrument users in an unobtrusive fashion.
- Assume the state of the user is observable through sensors.
System Design: Compute Subgoals

- Repeating raw observations verbatim can lead to several problems.
- Segmenting observations into symbols ignores uncertainty.
- Use likelihood of predictive model to determine subgoals.
System Design: Associate Subgoals to Environment

- Most tasks are defined with respect to objects in the environment.
- Associate subgoals automatically with objects in the environment.
- Potential problems: object occlusion, incorrect associations.
System Design: Another Demo?

- The user can demonstrate the task as many times as desired.
System Design: Map Demos to Common Environment

- Environment configuration may be different for each demonstration and cannot use object tracking.
- Map demonstrations to a “canonical” environment to minimize configuration-specific user behavior.
System Design: Map Demos to Common Environment

- Determining the mapping is an optimization problem, solutions depend on the objective function used.
- Implicit assumptions about likely perturbations are built into the objective function.
- Must be able to map environments with extraneous and occluded objects.
System Design: Determine Task Structure

- Combine observations from all demonstrations to determine the *intent* of the user, captured by an FSA.
- FSA output should be necessary set of subgoals to complete the task, including branching.
System Design: Perform Task

- Map canonical environment to current environment.
  - Mapping must work in both directions.

- Used necessary subgoals from FSA to perform the task.
System Design: User Modification?

- Integrate the user into the design loop to improve LBO system performance.
- Modifications should alter the way that the task structure is determined.
System Design: Sample Implementation

- Simulator description.
- Computing subgoals.
- Map demonstrations to common environment configuration.
- Determine Task Structure
- Performance Metric
Sample Implementation: Simulator

- Tasks consist of click-and-drag operations with the mouse
  - Mouse state is given directly by simulator.

- Environment is comprised of planar polygons.
  - Configuration consists of corner locations, found by vision algorithms.
Determining Subgoals

- Assume user follows smooth trajectories between subgoals.
- Estimate the parameters of a time-varying linear system using a moving average.
- When likelihood of predicting next state drops dramatically, mark previous state as subgoal.
Determining Mapping

- Attach springs from each corner to all others.
- Repeat for resultant frame.
- Minimize change in force for all objects.
  Weighted bipartite graph matching: Linear Program.
- Gets confused easily…
Sample Task

- First, find corners.
- Next, observe user and determine subgoals.
- Determine subgoal-environment associations.

- Asked four hapless grad students to demonstrate the task five times.
Sample Task

- Resulting subgoals determined from 20 user demonstrations.
- From these data, we must determine the structure of the task.
- Description of training algorithms and performance comparison.
Acyclic Probabilistic Finite Automaton Training Algorithm

- Each demonstration is an ordered set of subgoals.
- Build a DAG that is comprised of all demonstrations.
- Treat demonstrations as stochastic to combine similar subgoals.
Sample Task

- Output of APFA algorithm.
- Appears to capture intent of users well.

- But what does *well* mean?
Quantifying **Well**

- What’s missing from the diagram is a way of measuring which is better.

- Preliminary answer: Normalized string edit distance.
  - Percentage of subgoals that must be added, deleted, or modified to get “correct” answer.
    - APFA algorithm: 0.11 😊
    - HMM algorithm: 0.95 😞

- These are scores for canonical environment, not ensemble averages.
Proposed Work

- Determine “ergodic” performance metric of LBO system.
- Incorporate robust mapping algorithm that can handle object occlusion.
- Develop theory behind APFA-training algorithm.
- Incorporate users into design loop of LBO system.
- Integration of vision algorithms.
Evaluation of Research

• Cross-validation sets will be used on hand-crafted problems to evaluate subsystems.

• Target task is arc welding.

• Feedback from domain experts will be used to evaluate viability of system.
Expected Contributions

- Demonstrate that an LBO system is a viable automation tool.
- Create an LBO system resilient to environment perturbations.
- Determining subgoals in a non-symbolic fashion.
- Incorporating multiple demonstrations to increase performance.
- Devising an algorithm to determine the structure of underlying tasks.
# Preliminary Timetable

<table>
<thead>
<tr>
<th>Task</th>
<th>Start date</th>
<th>End date</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification of algorithms in simulation.</td>
<td>May 01</td>
<td>August 01</td>
<td>4 months</td>
</tr>
<tr>
<td>Integration of vision algorithms.</td>
<td>September 01</td>
<td>November 01</td>
<td>3 months</td>
</tr>
<tr>
<td>Controlled experiments on robot manipulators.</td>
<td>December 01</td>
<td>March 02</td>
<td>4 months</td>
</tr>
<tr>
<td>Experiments with domain experts and integrating their feedback</td>
<td>April 02</td>
<td>July 02</td>
<td>4 months</td>
</tr>
<tr>
<td>Write report</td>
<td>July 02</td>
<td>September 02</td>
<td>3 months</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>May 01</td>
<td>September 02</td>
<td>18 months</td>
</tr>
</tbody>
</table>