Annotation and analysis of video material from user study

Find specific actions and sequences of actions in video material recorded in a user study on high-level programming

Annotate the video material; identify reoccurring patterns in the material (proj 1) or identify significant situations indicating that a certain prompt is appropriate (proj 2).

Determine suitable segmentation of the video material
Determine suitable features for annotations / attributes in Bayesian Classifier for general patterns (proj 1)
Determine suitable features for attributes in Bayesian Classifier for prompt opportunities (proj 2)

[Make use of existing implementation of a parsing module to train a BN to identify such patterns]

Material:
  Video material
  ELAN annotation tool
  Software (ROS based) for parsing ELAN-output for training and use of BN

Contact:
  Elin_Anna.Topp@cs.lth.se
Reasoning about (geometric) constraints for end-user specified re-usable robot skills

Understand the (geometric) relationships of different relevant reference coordinate frames needed in a robot skill (e.g., pick and place of an object) to reason about possible (and impossible) transformation or transfer of the skill; determine configuration / relationships after a transfer from, e.g., one robot to another.

Make use of the skill representation used in the RSS group’s KIF
Extend the iconic programming interface currently used to ask for user input where necessary
Implement the geometric transformation(s) for the respective actions in a skill under given constraints either directly on the KIF-side or as part of the interface implementation

[Run experiments in a setup similar to the “SARAFun testbed setup”]

Material:
- Programming interface (C# on Windows 10)
- Access to KIF (RDF triple store)
- “AIMA”-book (Russel / Norvig) + other literature

Contact:
Elin_Anna.Topp@cs.lth.se
Dialogue (graphical pop-up) for user support in robot programming

*Use results from underlying reasoning mechanisms to formulate dialogue elements for the GUI used for our robot programming work.*

Extend the iconic programming interface currently used to ask for user input where necessary based on already existing dialogue elements.

Investigate options for more proactive dialogue based handling of situations where the user might need support.

[Run experiments in a setup similar to the “SARAFun testbed setup”]

**Material:**

- Programming interface (C# on Windows 10)
- Access to KIF (RDF triple store)
- Access to previous results (dialogue management in ROS / python and dialogue GUI for knowledge gap filling)
- “AIMA”-book (Russel / Norvig) + other literature

**Contact:**

Elin_Anna.Topp@cs.lth.se
Integration of RGB-D based object localisation and identification into robot programming tool

**Fully integrate results from an RGB-D camera based tool for object identification and localisation based on next best view heuristics into the programming tool. Support the user in controlling the robot towards the next best view or control the robot directly; integrate representation of results from vision system into the world model in the programming interface.**

Understand the interfaces between KIF and GUI / KIF and the RGB-D camera based object object identification.

Extend the GUI to integrate the results from object localisation and identification directly and thus support the user in the programming process.

[Run experiments in a setup similar to the “SARAFun testbed setup”]

Material:

- Programming interface (C# on Windows 10)
- Access to KIF (RDF triple store)
- Access to previous results (MSc thesis / EDAN70 project)
- “AIMA”-book (Russel / Norvig) + other literature

Contact:

Elin_Anna.Topp@cs.lth.se
Solving (robot) learning problems in OpenAI Gym

The OpenAI Gym features environments with robot arms and grippers to work on Reinforcement Learning approaches that solve, e.g., a cube positioning problem or a pick-and-place-task. The interesting aspect of this is that the environments give sparse rewards and have a specified goal (getting the object placed correctly, getting the cube into a specified pose).

Understand the interface from Python / TensorFlow to the chosen gym environment. Implement suitable algorithm(s) (check also the available implementation of HER in the gym). Report on thoughts regarding transfer options to the real world (we might even consider using one of our robots to do the pick-and-place task).

Material:

- Access to suitable machine for OpenAI Gym, Python 3.5
- (at least) and TensorFlow
- “AIMA”-book (Russel / Norvig) + other literature

Contact:

Elin_Anna.Topp@cs.lth.se
Solving (robot) learning problems in OpenAI Gym

The OpenAI Gym features classic environments (walker, lunar lander, mountain car, etc) to work on Reinforcement Learning approaches. The environments give shaped rewards and usually have vaguely formulated goals (land the lunar lander somewhere in a given area, walk forward, reach the mountain top).

Understand the interface from Python / TensorFlow to the chosen gym environment. Implement suitable algorithm(s) and evaluate (extend the last EDAN95 course assignment, if you will). Report on thoughts regarding transfer options to the real world (we might even consider using one of our robots to do, e.g., the “cartpole” task).

Material:
- Access to suitable machine for OpenAI Gym, Python 3.5
- (at least) and TensorFlow
- “AIMA”-book (Russel / Norvig) + other literature

Contact:
Elin_Anna.Topp@cs.lth.se