RoboEarth Web-Enabled and Knowledge-Based Active Perception

Web-enabled robots and RoboEarth

- RoboEarth: A set of servers, a joint database and a network that allow robots to:
  - Upload information.
  - Download information.
  - Outsource computation.
- EU-funded 4-year project, finishing this year.
Action Recipes for Active Perception Tasks

- Specify a task on a platform- and environment-independent level
- Formulated on an abstract level (Web Ontology Language OWL) that makes possible to share recipes among different robots.
- Due to the level of abstraction, is not viable to be used as input to an interpreter for task execution.

**Semantic Mapping Action Recipe**

- `loadTypiqueObjectModels`
  - **type**: LoadTypiqueObjectModels
  - **modelsForPlace**: HospitalRoom

- `startExplorationModules`
  - **type**: StartExplorationModules

- `exploreEnvironment`
  - **type**: re_explore

- `checkFinished`
  - **type**: CheckingWhetherConditionObtains
    - **objectActedOn**: re_explore
    - **conditionEvaluated**: COMPLETED_OK

- `stopExplorationModules`
  - **type**: StopExplorationModules

- `uploadVslamMap`
  - **type**: UploadMapToRobotEarth
    - **serviceCalled**: RevSLAMMapUploadService

**Object Search Action Recipe**

- `startVisionModules`
  - **type**: StartVisionModules

- `inferLikelyLocations`
  - **type**: InferLikelyObjectLocations
    - **objectActedOn**: Bottle

- `goToNextBestPose`
  - **type**: TranslationLocationChange
    - **nextBestPose**
      - **providedByMotionPrimitive**: navigate

- `checkObjectFound`
  - **type**: CheckingWhetherConditionObtains
    - **conditionEvaluated**: __Description20
The system has a set of manually written functions that generate partial execution plans for different OWL action concepts.

The code generated functions are stored in the RoboEarth database and annotated with an OWL description.

For each action described on the recipe, the system looks up possible code generator functions in the RoboEarth database and select the most specific one.

```lisp
(def-cram-function exploration
  (prog ()
    ...
    
    check-finished
    ...
    (cond (condition-true
            (go stop-exploration))
          (condition-false
           (go explore-environment)))
    ...)
```
Robot Capabilities for Active Perception Tasks

Robot Components

EnvironmentExplorationComponent
- 2D range finder frontier exploration
- Gmapping, robot localization
- A* algorithm, trajectory planning
- ORM, obstacle avoidance

ObjectRecogntinioinComponent
- SURF, object recognition
- PnP+RANSAC, initial object location

VisualSLAMComponent
- Keyframe bundle adjustment visual SLAM
- Distributed, in the cloud back-end implementation
- Recognized object insertion
My capabilities are ... How should I explore this room?
For this room, you have to explore according to this tailored for you CRAM plan, while looking for these two objects.
Semantic Mapping
Action Recipe

load-typical-object-models
\[\text{type: LoadTypicalObjectModels} \]
\[\text{modelsForPlace: HospitalRoom} \]

start-exploration-modules
\[\text{type: StartExplorationModules} \]

explore-environment
\[\text{type: re_explore} \]

check-finished
\[\text{type: CheckingWhetherConditionObtains} \]
\[\text{objectActedOn: re_explore} \]
\[\text{conditionEvaluated: COMPLETED_OK} \]

stop-exploration-modules
\[\text{type: StopExplorationModules} \]

upload-vslam-map
\[\text{type: UploadMapToRoboEarth} \]
\[\text{serviceCalled: REvSLAMMapUploadService} \]
Representation of maps as RoboEarth ontology environments

Environments generated:

- **SemanticEnvironmentMap**: reasoning.
- **Visual Map**: Object detection & location, semantic map grounding.
- **OctoMap**: 3D Occupancy grip map: Laser location, motion planning.
Object Search Action Recipe

My capabilities are ... . How should I find a drink?
For this room, you have to explore these selected locations according to the tailored for you CRAM plan. Search for these object models.
Object Search Action Recipe

start-vision-modules
  type: StartVisionModules

infer-likely-locations
  type: InferLikelyObjectLocations
  objectActedOn: Bottle

go-to-next-best-pose
  type: TranslationLocationChange
  dependsOnCapability: BaseMotionCapability
  toLocation: nextBestPose
  providedByMotionPrimitive: navigate

check-if-object-found
  type: CheckingWhetherConditionObtains
  conditionEvaluated: __Description20

stop-vision-modules
  type: StopVisionModules

upload-vslam-map
  type: UploadMapToRoboEarth
  serviceCalled: REvSLAMMapUploadService
 Commonsense Reasoning about Object Locations

- Probabilistic location models bootstrapped from OMICS [Gupta 2004]
  - OMICS's DB holds tuples of locations and proximity of objects: (objA, objB)
- Given object type A, what is the probability to encounter obj. type B?
  - \( P(B|A) = \frac{\text{count}(A,B)}{\text{count}(B)} \), counts DB entries
- Finally, retrieve known instances of object type B from the semantic map, go there, and search for object type A
- Example query:
  - ?- objLoc(?Loc,'Bottle').
  - Loc = [0.16,'Cabinet'].

Commonsense Reasoning about Object Locations
Reasoning about Object Visibility

- Searching for objects requires
  - Compute likely object positions
  - Compute robot pose from where that position is visible

- Visibility computation based on semantic environment map
  - OpenGL rendering from object position to find occlusions
  - Costmap of visibility from robot positions based on robot model
Environment Exploration Component

Goal: Safely navigation and mapping in an unknown environment

- Gmapping approach for localization and mapping.
- Based on a frontier laser exploration.
- Trajectory planning based on the A* algorithm.
- ORM obstacle avoidance method.
Object recognition Component

RoboEarth provides a tailored object database (~100) for each task

**Goal:** Reliable recognition at a very low execution time:
- Loops detection: 22 ms per image (database size ~26000 images).
- Object recognition and insertion: 28 ms per image (~20 3D objects).
- State of the art SURF-based recognition: ~300 ms per image
Visual SLAM Component

**Goal:** Sparse map for localization purposes, object registration, and point cloud registration.

- Proposal: An architecture C2TAM based on PTAM – Parallel Tracking and Mapping.
- Achievements:
  - Data flow within standard Wi-Fi
  - Cooperative mapping.
- Advantages:
  - Outsourcing computation.
  - Potential for massive storage of annotated data.
  - Low cost front end.
- Outputs:
  - **SemanticEnvironmentMap**: using the object recognition component.
  - **Octomap**: 3D Occupancy grid map.
  - **VSLAMmap**: Features map for localization.
Real-world experiments

- A robot in a hospital room has to find a bottle to be served to a patient.
- The proposed method exploit a knowledge-based search strategy based on a action recipe.
  - *SemanticMapping* recipe enables the robot to efficiently create a semantic map during the exploration.
  - *ObjectSearch recipe* allows guided search for objects based on a partial semantic map of the environment. The object location inference and visibility reasoning determine the likely locations for searching the object.
Simulation Experiments & Performance Improvements

Simulation Experiments

- Show the applicability of the system on:
  - Different robots.
  - Different environments.
- For the visibility reasoning, the pose of the camera was inferred from the SRDL descriptions.

Performance Improvements

- Show the improvement made by the proposed system doing the same experiments without using the information of the semantic map.
- Performing and exhaustive search we need 20-100 position for full coverage the entire environment.
- Using the presented inference method we are able to reduce the number of positions to 9-15.
Conclusions

- A basic robot with state-of-art navigation and perception capabilities is not able to efficiently explore and actively search for an object. The robot performance in active perception is boosted by web services provided by RoboEarth.
- A customized execution plan is created for the current robot and the current environment.
- RoboEarth provides the robots with a selection of only relevant models for the current task. It allows to obtain perception with high precision and recall in real time.