Collaborative R&D program:

Ambient Assisted Living services and skills emulation with personal robots

A. Project Identification

Project Title:

Natural-like human-personal robot interaction system for AAL services: communication and skills learning through visual gestures interpretation

Project Description:

The proposal aims at developing a software service interface for natural-like interaction of humans with PR2-type personal robots. Natural-like interaction will be considered for communication (a model of hand gestures will be established that allow many natural gestures to be interpreted by the personal robot) and emulation of human skills, routine tasks (extracting reusable task knowledge from visual observation of human gestures, learning operational activities from human demonstration). The proposal involves a number of sections which will be integrated in this software system: (i) Spatial and temporal modelling of communicative and manipulative gestures; (ii) Hand gestures analysis and recognition, based on multiple-image processing); (iii) Reusable task knowledge extractor from visual observation of human performance and action reproducer (human tasks emulator); (iv) Visual servoing controller for environmental motion fine tuning. The OpenCV library will be used for image processing; an open source library will be created for gesture modelling, analysis and recognition.

Principal investigator:

Professor Theodor Borangiu, Ph.D., theodor.borangiu@cimr.pub.ro

Investigators:

Professor Adina Florea, Ph.D.
Professor Dan Popescu, Ph.D.
Associate Professor Andrei-Nick Ivanescu, Ph.D.
Associate Professor Florin Anton, Ph.D.
Lecturer Silviu Raileanu, Ph.D.
Assistant Professor Iulia Iacob, Ph.D.

Institution Name:

University Politehnica of Bucharest, Faculty of Control and Computers – develops the solution
Company Partner: Willow Garage, USA – provides the PR2 robot

B. Project Summary

This project attempts to develop a natural human-robot interface for programming and operating in order to make personal robots more accessible for services to people without a vast technical experience, allowing the robots to assist the humans in everyday tasks. The project consists of four sections, which focus on vision-based hand gesture modelling, analysis and recognition, robot programming by demonstration, automatic learning of advanced skills from the human operator and visual servo control for fine adjustments of the robot motion to a continuously-changing unstructured environment. An overview for the project components is given in Fig. 1.

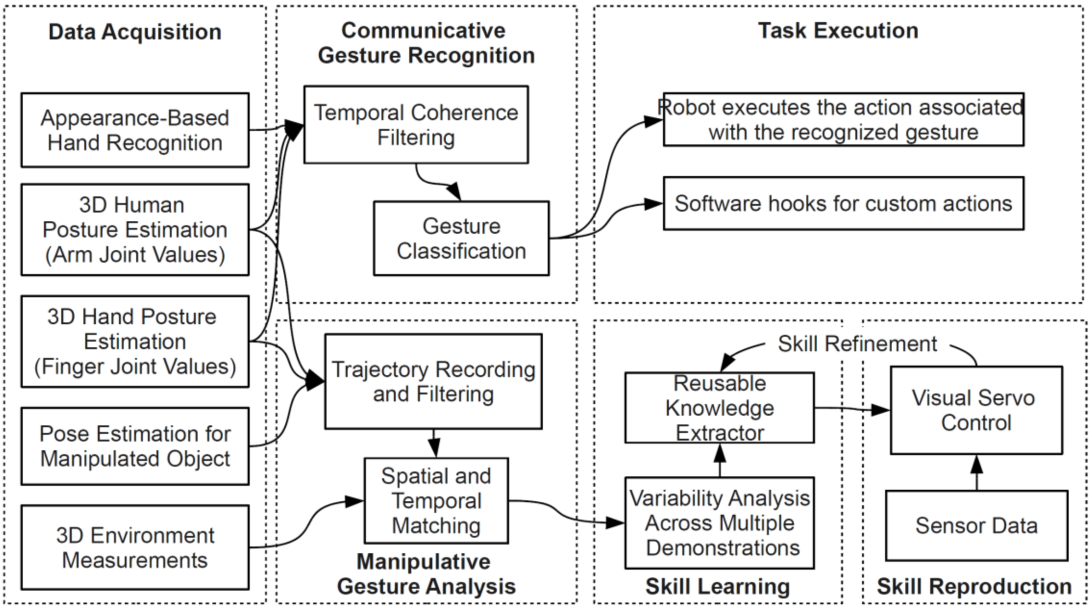

Fig. 1. Overview of the proposed software components

B1. Spatial and temporal modeling of communicative and manipulative gestures

This section aims to develop a set of mathematical representations for human gestures, in order to facilitate the process of recognition and/or imitation.

Fig. 2. Communicative and manipulative gestures to be used for PR2 control

After separation from unintentional movements, human gestures can be divided in two main classes: *manipulative* and *communicative* (Fig. 2). Manipulative gestures are the ones when the user performs an action on objects in the environment, which can be imitated later by the robot or can be used to teach new skills. On the other hand, communicative gestures may represent commands given to the robot, like “Stop”, “Follow Me”, “Faster” or “Slower”, may indicate the object of interest (pointing with the finger or gazing) or may indicate approval or disapproval (yes/no) which is useful in the learning process.

From a recognition point of view, gestures can be classified in *pose gestures* (static configurations of the arm or hand) or *motion gestures*, which are defined by specific motion patterns of the arm or hand. However, static gestures can be easily mismatched especially in cluttered backgrounds; it is not desirable for a robot to refuse to work just because there is a picture on the wall with a hand symbolizing a “Stop” gesture.

B2. Hand gesture analysis and recognition, based on multiple image processing

The problem of hand segmentation and gesture recognition in 2D images has been studied extensively; however, a single 2D camera does not provide depth information, therefore it is only suitable for communicative gestures.

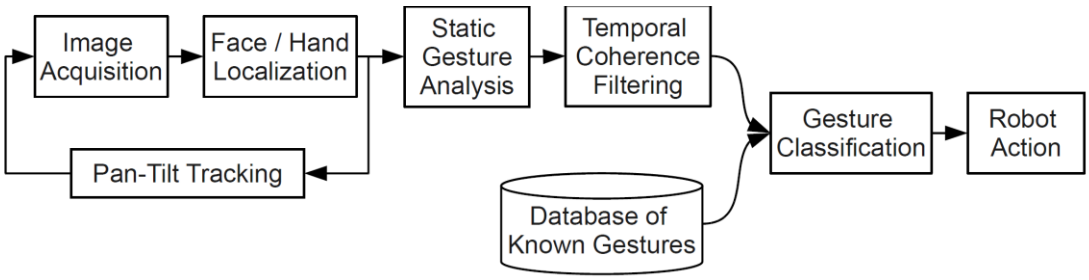

Fig. 3. Proposed communicative gesture recognition loop

Since the PR2 is equipped with two stereo camera pairs, with both wide and narrow viewing angles, and a pan-tilt unit, a software module for tracking the required features (in our case, the human hand) will be developed.

For improved robustness, all the gestures (including static ones) can be considered to be composed of three phases, therefore regarded only as motion gestures:

- preparation (moving the arm from some resting position)
- stroke (the effective motion pattern which defines the gesture)
- retraction (return to the resting position or prepare for a new gesture)

For communicative gestures, it is desired that the robot reacts as soon as the user intent is recognized; therefore, for certain gestures, such as the “Stop” command, the robot should stop immediately, not after a few seconds or after waiting for the retraction phase to finish.

The first step in gesture recognition is locating the hand in the input images. Appearance based models, like Haar classifiers or eigenfeatures, can be employed for an initial guess; however, both face hand localization algorithms in a single image can give false positives, therefore it is necessary to also consider the temporal coherence of the hand and face location throughout a sequence of images.

Recognizing a communicative gesture implies comparing the gesture acquired by the robot sensors against a database of known (previously learned) gestures. This can be accomplished using a metric function.

The following difficulties have to be addressed when designing the metric function for comparing a gesture acquired by robot with the models stored in the database:

- The gestures may differ slightly in time, both globally (the total duration) or locally (speed varies through the gesture execution). Dynamic Time Warping is a technique which improves matching quality in this aspect.
- The same gesture can be described by more than one valid trajectory in the parameter space, therefore statistical techniques are expected to give good results
- The dimensionality of parameter space is very high; therefore automatic selection of the features which can discriminate between gestures is desired. Techniques for accomplishing this are Principal Component Analysis and Cluster Analysis.

A manipulative gesture is defined by the following spatial and temporal information, which are obtained using the software model from Fig. 3:

- The 3D position and orientation of the manipulated object
- The joint angles of the human arm
- Location of the grasping points on the manipulated object

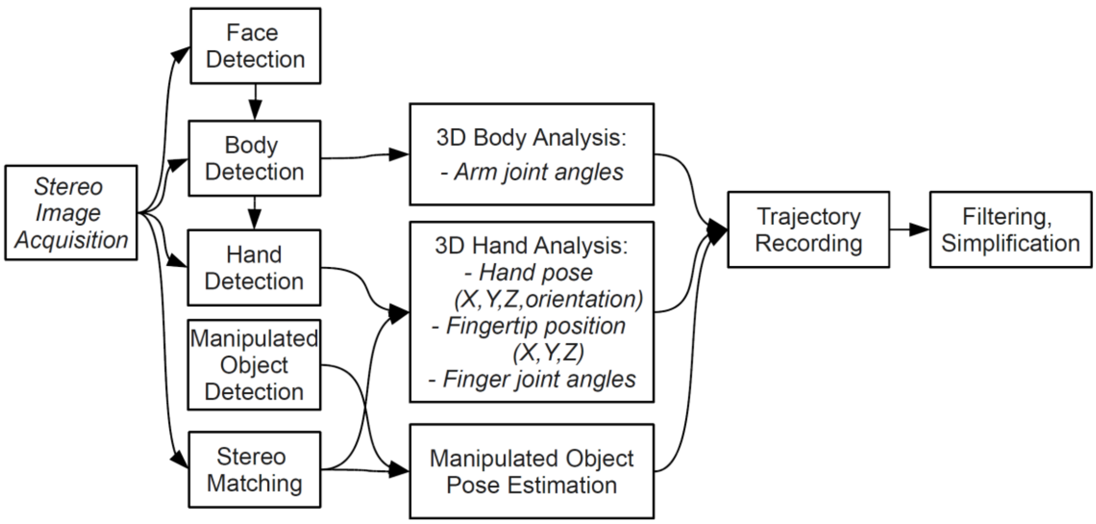

Fig. 3. Proposed model for manipulative gesture analysis

Since the PR2 is also equipped with a laser scanning sensor, the data from the stereo camera will be fused with the data from the laser scanner in order to obtain a more complete and accurate 3D data and for improved robustness.

Software libraries developed in this section include:

- module for 3D data fusion from stereo camera and laser scanner
- module for pan-tilt tracking of arbitrary objects, optimized for smooth hand tracking
- module for joint angle estimation of body segments and fingers from stereo images
- module for pose estimation of objects manipulated by the human arm
- module for gesture analysis, classification and recognition

B3. Reusable task knowledge extractor from visual observation of human performance and action reproducer (human tasks emulator)

Personal robots assisting humans can become widespread only if they will be easy to program by people without a technical background. This project attempts to teach the robot new skills in an intuitive way, by letting the robot watch human demonstrator doing the task.

The simplest approach is to follow the trajectory of the human demonstrator blindly, without any knowledge of the task. A more general approach is how to teach the robot to generalize the learned skill when the environment is dynamic, which is possible by using a higher level representation of the skill, which allows generalization in new environments.

It is possible to learn a skill from watching a single demonstration performed by the human teacher; however, the robot will have no means to know which components from the skill are important to reproduce exactly and which components can be varied in order to generalize the skill to a new environment. However, for simple skills, watching a single demonstration may be sufficient.

Reusable knowledge can be achieved by watching multiple demonstrations (however, as few as possible) of the human teacher and analysing the patterns which were common across all demonstrations, and also the patterns which were different. This process exploits the task variability across the demonstrations in order to detect which parts of the skill should be reproduced exactly and which parts do not.

B4. Visual servo controller for environmental motion fine tuning

The representative *dynamic look-and-move* system architecture will be analysed and implemented (open loop robot-vision architecture Fig 2.) The generic task which will be analysed is visual tracking of objects for robot grasping; solutions will be developed for tracking and grasping stationary objects. A number of original, low time consuming algorithms will be implemented and exemplified in case studies for: camera-robot calibration, evaluation of TOOL transformations using the robot itself as measuring device, robot-object calibration, modeling the grasping style and checking at run time for collision-free object grasping.

Visual servo involves *controlling* the motion to manipulate the robot environment using vision, as opposed to just *observing* the environment, like in active vision and structure from motion.

Visual servoing of robots uses structural features extracted from images as *form* – and *contour image features* for object recognition and locating or collision avoidance, *context features* may be added to this data to simplify object search at run time.

The form and contour image features refer to the projection of a *body* – or *hole* physical feature of an object (e.g. the part to be grasped, the gripper's fingers or the robot tool) onto the virtual camera image plane. Typical image features are: *edges* and *corners* for contours, respectively the *shape*, *centre of mass*, *orientation* of bodies and holes or *contrived patterns* for form descriptors. Image features must be unambiguously located in different views of the robot scene by different virtual cameras. *Image feature parameters* represent real-valued quantities that are computed from one or several image features.

Hierarchical motion control structure, with the vision processor providing set-points as reference input to the robot's joint-level controller – thus using joint data feedback to internally stabilize the robot. This structure corresponds to the *interlaced look-and-move* control schemes represented in Fig. 4, where motion tracking and image processing are *pipelined*:

- while the execution of an individual motion segment proceeds, no image is acquired and processed, and
- while an image is taken and treated according to the specific needs of a robot task, the motion controller does not start generating and tracking a trajectory

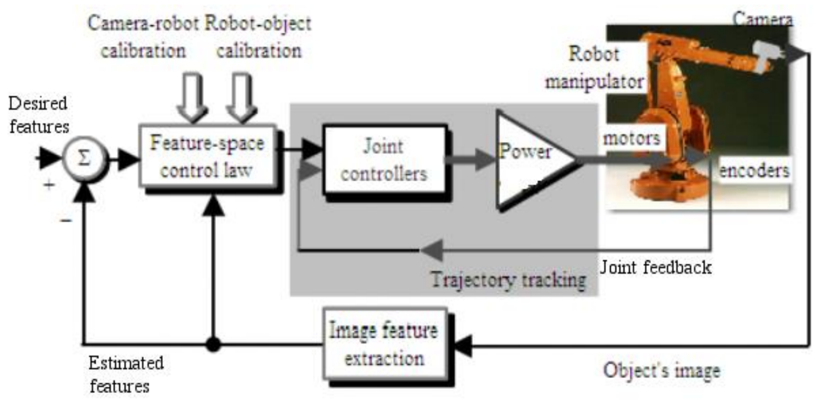

Fig. 4. Image-based look-and-move servoing architecture for object tracking

Image-based control structures, represented in Fig. 1 use error signals defined directly in terms of image features extracted from binary or grey scale object's images. In image-based servoing, control values are computed at run time directly from image features, which significantly reduces image interpretation and eliminates errors due to sensor modelling and camera calibration. However it does induce an important challenge to motion controller design, since the manipulator plant is nonlinear and strongly coupled.

The expected results of this sub-theme of research will be materialized in the following libraries and databases:

- An object model database used for visual recognition
- A library used to extract different features from images
- A library used for visual servoing

C. Funding

Current funds:

Autonomous, SOA integrated, self-learning adaptive robot-CNC control system for reverse engineering of complex-shape objects from depth map images
Grant 69 / 10.01.2009 – 10.01.2011
Funding agency: National Council for Scientific Research in Higher Education (CNCSIS)

Ambient Assisted Living, assisting aged people in home tasks AmiCare Project, 2011 – 2013
Funding agency: Ministry of National Education (MEN)

Necessary budget:
200 K euro, 2 years for: development, implementation, testing and validation, dissemination

D. Open Source Community Plans

The libraries to be released for this project will contain the following functionality:

- Modeling, analysis, learning and recognition of communicative hand gestures
- Pan-Tilt tracking of arbitrary objects using the wide and narrow angle stereo cameras mounted on the PR2 head
- Data fusion for 3D information from laser scanning and stereo vision
- Joint angle estimation of human body segments and fingers from stereo vision
- Pose estimation of objects manipulated by the human arm
- Task emulation, knowledge extraction and generalization functions from visual observation of human performance
- An object model database used for visual recognition
- A library used to extract different features from images
- A visual servo control library for the PR2

Also, any functionality which may be needed during the project and are not yet implemented into ROS or available as ROS packages will be released as open source libraries.

We plan to update the code on the project on monthly basis.

E. Intellectual Property

The libraries for this project will be released under GNU GPL and BSD license and will not rely on any proprietary libraries or source code. There are no IP constraints on this project and no contractual barriers to face in making code from this project open source.

Development environments and libraries considered for use are primarily Octave, Python, OpenCV. Additional open source libraries to be considered are:

- TINA (3D machine vision toolkit for robotics and medical use)
- *lbimproved* (Dynamic Time Warping implementation)
- GVF (Gradient Vector Flow – improved Active Contours / Snakes)
- AAM-API (Active Appearance Models for flexible/deformable object tracking)
- ITK (Toolkit for image segmentation and registration; designed for medical use)
- HandVu and visualgestures (Hand gesture recognition and tracking from single 2D camera; both projects use OpenCV)