Integration of the Humanoid Robot Nao inside a Smart Home: A Case Study

Athanasia Louloudi, Ahmed Mosallam, Naresh Marturi, Pieter Janse and Victor Hernandez

School of Science and Technology
Örebro University, Sweden 701 82
`<first name>.<last name>h081@student.oru.se`

April 26, 2010

Abstract

This paper presents a case study demonstrating the integration of the humanoid robotic platform Nao within a Network Robot System (NRS) application. The specific scenario of interest takes place in a smart home environment; the task being that of bringing a can of soda from a fridge to a human user. We use this concrete scenario to evaluate how the performance of such a robot can be affected by being embedded inside an intelligent domestic environment. This study points out that, by cooperating with different components on the network the overall performance of the robot is increased.
Keywords: Network Robotics Systems, Domestic Robots, Nao robot, Smart home, PEIS-Ecology.

1 Introduction

A few years ago, the idea of living with robots, sharing everyday tasks and harmonically coexist in the same environment, seemed to be a distant scenario. However, the use of such technologies, aimed to inhabit our houses and help us with our everyday chores is not a dream any longer. Ongoing research all over the world indicates a trend to develop advanced robotic systems aimed to the service of people in need. According to the International Federation of Robotics, 7.1 million service robots for personal and private use were sold by the end of 2009 and 11.6 million is anticipated to be sold by 2012.[1] The rapid growth in the field of robotics during the last decade provided a strong foundation for smart homes and sensor networks.

A Smart home is a domestic intelligent environment where various components such as a fridge, oven, lights etc., are working together by exchanging information via the same local network. The principal idea behind the smart home concept is to use NRS techniques to integrate different services within the home in an effort to control and monitor the entire living space [1]. NRS can provide robot based services to improve care cost and the quality of life in smart homes. These services are not realized by a single stand-alone robot but by a combination of different elements such as environmental sensors, cameras, laser range scanners and humans communicating and cooperating through a network.

In a stand-alone robot, all the sensorial and computational capabilities are self contained. In the context of a NRS, a stand-alone robot is perceived as part of the ecology itself [2], [3]. Moreover, it can be beneficiated by the flux of information coming from other devices connected to the same network, for example, a

[1] *International Federation of Robotics*, Executive Summary of World Robotics 2009. See `http://www.ifr.org/service-robots/statistics/`

camera mounted to the ceiling can provide a wider view than its own embedded cameras. Network robots are divided into three types: visible robots, unconscious robots, and virtual robots [4], [5]. In this work we consider Nao as a visible robot since it has the role of being the physical interface of the NRS inside the smart home. Furthermore, according to Scopelliti et al., the most preferred robots, are those which are human-friendly in means of appearance, primarily resembling pets or toys [6].

This paper is organized as follows: First, in section 2, the problem formulation is presented, describing the overall tasks and the available tools that endue this work. Then, in section 3 we analyse some specific problems which arose in the implementation of our demonstration, emphasising in particular the contrast between our NRS approach and an alternative, single robot approach. The software architecture follows in section 4. Here, the methodologies and structure followed in order to handle this work are described. Finally concluding remarks close this paper.

2 Problem Formulation

Consider the following scenario:

Neils returns home from a long walk. Soon he enters the living room he taps on the head of Tommy, a humanoid robot and rests on the sofa. Tommy perceives the request, offers to bring a refreshment to Niels and starts moving towards the fridge in the kitchen. While walking, it localizes itself to find its position in the room. When Tommy enters the kitchen, it asks the fridge to open its door and use its gripper to collect a soda can and bring it out. The robot identifies the requested drink, grasps it and returns to deliver the refreshment to Niels.

The above scenario can be decomposed into the following simpler tasks that has to be performed by the robot in order to fulfill the overall goal:

1. Walk towards the fridge
2. Dock the fridge
3. Grasp the drink
4. Carry and hold the drink while walking
5. Deliver the drink to the user

These tasks can be grouped into three modules which are localization, cooperative grasping and the mobility module (explained in later sections of this paper).

The following paragraphs, introduce all the information needed in order to understand the components that surround this project. We describe the NRS infrastructure underlying our smart home, and other details that constitute the robotic platform Nao.

2.1 Available Resources

This section describes all the available tools that can be used in order to successfully accomplish the overall task.

2.1.1 Test Environment

PEIS-Ecology:
The concept of PEIS-Ecology, first introduced by Saffiotti and Broxvall in 2005 [7], is one of the few existing realizations of the notion of network robot system. The name PEIS stands for *physically embedded intelligent systems*. PEIS can be defined as a set of interconnected components, residing in one physical entity which generalizes the notion of robot. Every component that is part of this ecology is called PEIS-component.

The PEIS Ecology model has been implemented in an open source middleware, called the PEIS Kernel to which all the PEIS components are linked. This PEIS-Kernel allows the components to communicate and collaborate with each other in a standardized way. For communication, this middleware establishes a peer-to-peer network and performs dynamic routing of messages between PEIS. All PEIS can cooperate using a *uniform cooperation model*, based on the notion of linking functional components: each participating PEIS can use functionalities from other PEIS in the ecology in order to compensate or to complement its own.

PEIS Home:
The PEIS home[2] is an experimental environment that looks like a typical bachelor apartment which was built to implement the PEIS-Ecology [8]. It consists of a living hall, bedroom and a small kitchen. The PEIS-Home is equipped with communication and computational infrastructure. This study is conducted inside the smart home to make use of various components of PEIS-ecology. Fig. 1 illustrates a few basic views of the home.

The PEIS components available for this work are:

- PEIS-Fridge
  The PEIS-Fridge is a small sized refrigerator with a motorized door, a camera and an attached robotic arm with a gripper. The camera takes images over a shelf in the fridge and in combination with clustering algorithms directs the gripper towards the soda cans. The fridge gripper is able to collect the soda can from the interior space and is able to bring the drink outside.

- PEIS-Cam
  A PEIS-Cam is a component that can provide images from common 2D colour cameras. Several cameras are mounted in discrete areas of the house which can provide a wide view of the living space.

- PEIS-PersonTracking
  This component is a tracking system connected to a set of stereo camera and normal mega pixel camera mounted on the ceiling, capable of tracking multiple persons.

2.1.2 Nao robot

Nao is a humanoid robot equipped with sonar sensors, 2 CMOS cameras and three-fingered robotic hands. It features a multimedia system with 4 microphones and 2 hi-fi speakers for voice recognition and text-to-speech synthesis. The built-in functionalities of this robot such as logo detection, mark detection using onboard cameras are loftily adjustable and are used for robot localization. The robotic hands are used for grasping and holding small objects. Nao can carry up to 300g using both hands. [3]

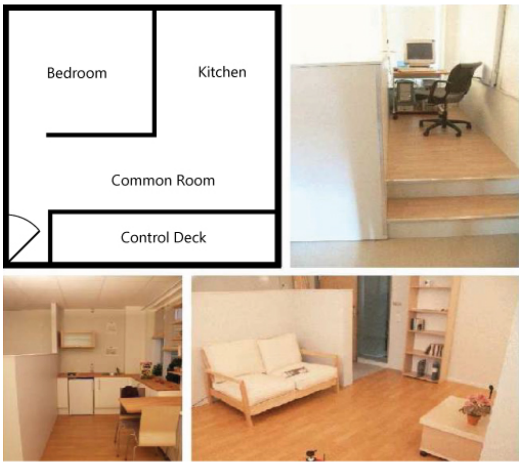

Figure 1: A rough sketch of the PEIS-Home*(upper left)*. A picture of the control deck *(upper right)*. A picture of the kitchen with a fridge located under the workbench *(lower left)*. A picture of the common room*(lower right)*.

Nao Robocup Edition has 21 degrees of freedom (DOF) whereas Nao Academics Edition has 25 DOF since it is built with two hands with gripping abilities. For this study the academic version of Nao is used.

The Nao is based on Linux and it is a fully programmable robot which uses its own framework called NaoQi. NaoQi, allows the developer to access all the features and functionalities of the robot through an Application Program Interface (API) which also provides the flexibility of executing tasks in sequential order, parallel and event based. The Integrated wireless network card of this robot can be used to exchange information with other devices in the network.

[2]The PEIS home is developed by the Center for Applied Autonomous Sensor Systems (AASS). See `http://aass.oru.se/~peis/`

[3]*Nao Documentation.* Aldebaran Robotics. 2009. See `http://academics.aldebaran-robotics.com`

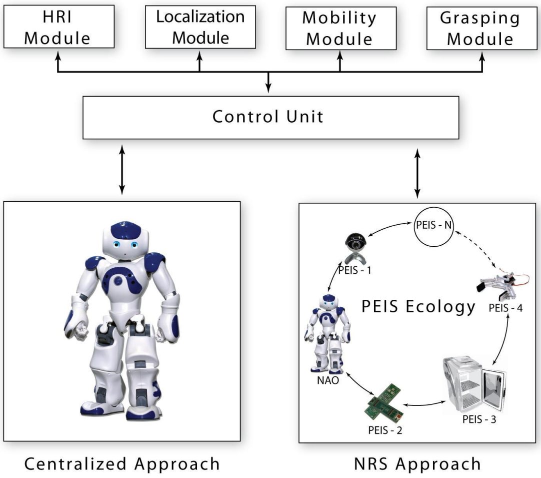

Figure 2: Software Architecture - Centralized and NRS approach

3 The Nao robot inside the PEIS-home

The first attempts to accomplish the scenario explained in section 2 were based on the robot's capabilities (centralized approach shown in Fig. 2), where all the perceptual information of the environment and the tasks are performed by one single agent. Considering that this humanoid platform is quite capable, it was expected to be able to perform successfully the scenario's tasks (localization, grasping and navigation) but several drawbacks revealed through testing. These drawbacks were the main reason for relying on external sources to fulfill the task. The first drawback has to do with localisation. The Nao is equipped with sonars, two cameras and a built-in logo recognition ("ALLandMarkDetection"), a vision module in which Nao recognizes special landmarks with specific patterns on them. Our attempts to leverage these sensory inputs for localisation failed, due to the following reasons:

1. The unreliability of the internal mark detection algorithm led to missing the checkpoint (marks) on the predefined path.
2. Because of the wobbling of Nao while walking the detection of the marks was unreliable.
3. Slight changes in the lighting conditions of the room, could lead to faulty detection or not detection at all.
4. State of the art localisation using stereo vision [12] cannot be implemented using the two onboard cameras since there is no overlapping region between the data acquired by them.

The following section, describes the final implementation. The software architecture will be described and a more detailed illustration of the basic three modules will ensue.

4 Software Architecture

The software architecture is shown in Fig. 2 and was built having in mind the NRS approach which; as explained in [5], has the premise of combining robotics and information communication technologies through a network in order to realize a task or provide services to human users. The software architecture was developed using Aldebaran's NaoQi as a framework and coded in C++. NaoQi gives access to all the features of the robot, like sending commands to the actuators, retrieving information from the robot memory, and managing Wi-Fi connections. This architecture facilates the integration of new components without redesigning the system architecture.

4.1 Control Unit

The control unit is built around a state machine, which provides the logic to arbitrate the interaction between the rest of the software modules and determines the sequence of the actions that should take place in order to achieve the goal described in the introduction of this paper.

Fig. 3 shows the state diagram of the control unit were each state represents an action to be executed and the transitions between states are determined by signals passed to the control unit by the rest of the modules.

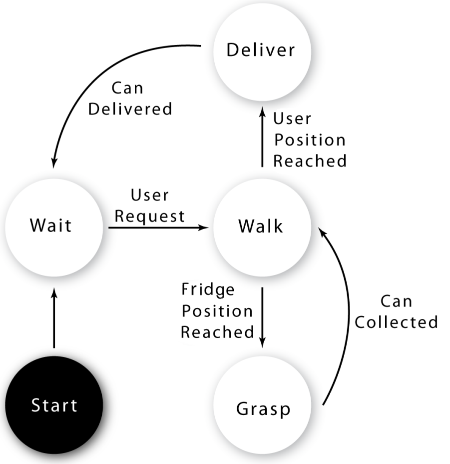

Figure 3: State diagram for the control unit

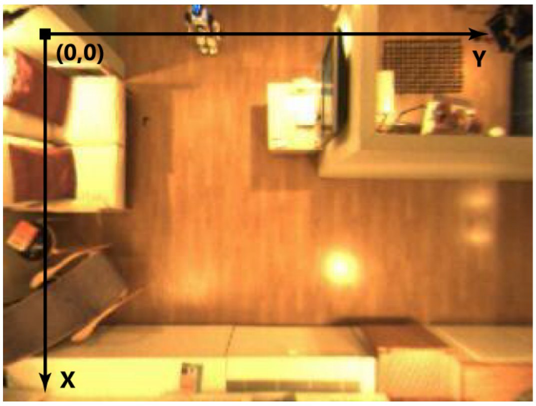

Figure 4: Coordinate frame for the mobility module

4.2 The Mobility Module

The function of this module is to handle the navigation of the Nao robot inside the PEIS home. This module receives as inputs the estimated current pose and the commanded poses through the controller module.

To represent the robots pose in the PEIS home, we have opted for the model suggested in [9], where the degrees of freedom of the joints of the legs and the feet of the robot are ignored and the robot is represented as a rigid body operating in a horizontal plane. The robot

Table 1: PSEUDO-CODE FOR MOBILITY MODULE

1	Retrieve the current pose from the localization module.
2	Compute the angle AT between the desired pose and the current pose.
3	Compute the Euclidian distance DS between the desired pose and the current pose.
4	**if** $AT > anglethreshold$ Command Nao to turn AT degrees.
5	**if** $(DS > Distancethreshold)$ Command Nao to move DS meters forward.
6	When Nao stops moving, repeat steps 1 to 5 to reduce the error in the pose.
7	Wait for the next command from the control unit.

pose(RP) is represented as follows:

$$\vec{\mathbf{RP}} = [x, y, \theta]^T \tag{1}$$

where x and y are the position in the coordinate frame shown in Fig. 4, and θ is the orientation of the robot with respect to the x axis.

The error between the current and the commanded pose is decomposed in an orientation error and a position error, that are later passed through the Nao API's to orient the robot towards the target position and then to move it forward to correct the position error. However, the built in humanoid walking implementation of the Nao robot does not consider the external disturbances that affect the walking schema of the robot. This lack of feedback causes an error between the desired pose and the physical pose of the robot. In order to reduce the impact of this error, the reported pose from the localization module is used to compensate the desired pose of the robot.

The functionality of the mobility module can be summarized in Table 1.

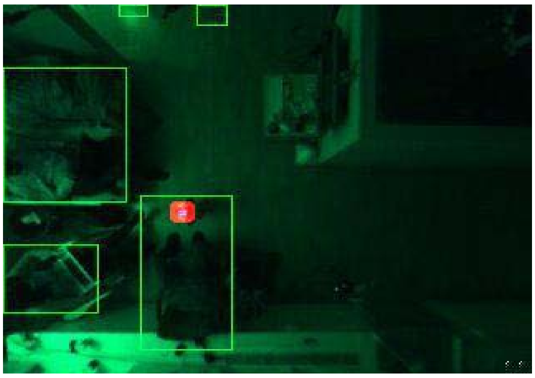

Figure 5: Robot detection and tracking using the background subtraction and color slicing algorithms

4.3 The Localization and Coordination Module

As mentioned before the odometry error between the desired and the actual pose of the robot is present due to external disturbances that affect the robot while walking. In order to reduce the impact of this error, we use an ad-hoc odometry algorithm that gives the robot a feedback about its actual position. Using already present environmental monitoring cameras placed on the ceiling, background subtraction [10] and color slicing algorithms [11], we could efficiently localize the robot by detecting the blue light on top of its head and provide the mobility module with the actual pose of the robot as shown in Fig. 5. We used this algorithm to calculate the robot position at already known positions and built a lookup table that we used later to interpolate the robot position. The pseudo code in Table 2, presents the basic steps of the algorithm. We did that to decrease the effect of the noisy readings of the robot position due to the robot instability while walking.

The Localization module can be divided into three steps:

1. Background subtraction
2. Color slicing
3. Position lookup table

Table 2: PSEUDO-CODE FOR LOCALIZATION AND COORDINATION MODULE

```
1  Capture background image B”
2  Subtract the new frame from B.
3  for each resulting region:
     if area > areathreshold
        add this region to ROI
4  for each ROI:
     for each pixel in the selected ROI
       Calculate the Euclidian distance
       between this pixel and the color
       threshold C”.
         if C > colorthreshold
           Set the pixel value to one
         else
           Set the pixel value to zero
5  The region with highest value will be
   the robot pose R”.
6  Calculate the centroid for R and the re-
   sult will be single pixel P”.
7  The lookup table calculates the robot
   position using P.
```

4.3.1 Background subtraction

In this step the algorithm subtracts each new image from the background image and detects the regions of interest (ROI). ROI area has to be greater than a certain threshold. The threshold has to be set manually according to the ”tolerance” we want to give to our algorithm (a high value of threshold means that only objects with high area value is considered interesting, whereas a lower value of threshold will make the algorithm detect small objects as ROI).

4.3.2 Color slicing

At this point, we consider only the ROI delimited in the previous step. The ROI is processed in the RGB color space, where we calculate the Euclidean distance between the pixel in the ROI and a color of interest. If the distance is less than a certain threshold, the pixel is classified as part of the robot.

4.3.3 Position lookup table

In this final step, we use a lookup table to return the real robot pose. This lookup table provides a mapping from the robot's position in pixels to the real coordinates.

Although this ad-hoc solution is quite simple, the results obtained have significantly reduced the error in the orientation in comparison with the open loop walking schema.

4.4 The Cooperative Grasping Module

The robotic grasping is a very difficult problem, specifically grasping objects that are being seen for the first time through vision. Solving this problem could be through using learning algorithm or building a 3-D model of the object of interests or by using both techniques. In this work we did not apply any learning or modeling approaches, instead we assumed that a specific object is handed in to the robot in a fixed pose. That means whenever the robot reaches the right position for grasping it will start predefined movements in order to grasp with no feedback. The grasping problem was simplified in order to tackle the problems of detecting the presence of the object to be grasped and approaching it. We installed a light source on top of the fridge gripper to be detected in the same way as we detect the Nao robot head. The algorithm detects the robot and the object to be grasped and calculates the Euclidian distance between the robot and the object.

The walking module receives the distance and commands the robot to move. After approaching the grasping point, the robot does the following:

1. Command the fridge to open through the PEIS infrastructure
2. Command the fridge gripper to fetch a certain drink out of the fridge
3. The robot docks the extended fridge gripper to place itself in a position suitable for grasping. This is done by indirect visual servoing through the localisation of the gripper as perceived by the environmental camera
4. The robot then follows predefined arm and hand movements to grasp the object. See Fig. 6

Figure 6: Fridge door opening*(upper left)*, fridge gripper bringing the can out*(upper middle)*, tracked image from the camera *(upper right)*, Pre-defined robot arm movements after reaching the desired position for grasping the object*(bottom left, bottom middle and bottom right)*

The algorithm is explained in Table 3.

4.5 HRI Module

The Human Robot Interface (HRI) module acts as the direct link between the user and the system. Through this module the user is able to request the can of soda by tapping the robots head.

This module will receive the current position (coordinates) of the human user from the PEIS person tracking system. These coordinates in turn will be passed to the mobility module through the control unit as a final destination for the Nao robot.

The position of the user is determined by this module. See Fig. 2. When a centralized approach is used, the task of finding the user is performed by the robot itself (i.e. face recognition), while for the NRS approach, this task is carried out by an external module (i.e. the PEIS person tracker).

Table 3: PSEUDO-CODE FOR COOPERATIVE GRASPING MODULE

```
1   Capture background image B"
2   Subtract the new frame from B.
3   for each resulting region:
        if area > areathreshold
            add this region to ROI
4   for each ROI:
        for each pixel in the selected ROI
        Calculate the Euclidian distance
        between this pixel and the color
        thresholds C1 and C2.
            if C1 > colorthreshold
                set the pixel value to 1 for the
                robot region RROI
            else if C2 > colorthreshold
                Set the pixel value to 1 for the
                object region OROI
            else
                Set the pixel value to 0.
5   In RROI the region with highest value
    will be the robot pose R.
6   In OROI the region with highest value
    will be the object pose O.
7   Calculate the centroid for R and the re-
    sult will be single pixel "PR"
8   Calculate the centroid for R and the re-
    sult will be single pixel "PO"
9   The lookup table calculates the robot
    position and the object position using
    PR and PO.
10  Calculate the Euclidian distance and
    command the robot to approach this
    position.
    if (Destinationreached)
        Start the predefined movements of
        the arms.
```

5 System Demonstration

The system was evaluated using two different setups where, inspired by the scenario depicted in section 2, the robot was commanded by the user to follow a predefined path (from the living room to the kitchen) in order to collect and bring back a known object placed in the fridge gripper. Twenty rounds for each setup were performed.

In the first setup, it was required to make the robot locate itself inside the environment without being assisted by any other component inside the PEIS home. We opted for using the on-board vision capabilities of the robot to detect pre defined marks (i.e. Nao Landmark Detection) as shown in Fig. 7 and the marks were placed along the predefined path from the living room to the kitchen and the robot was expected to use them to correct its position and orientation in order to complete one test round. Poor results were obtained due to the noise introduced from embedding the localization module inside one single agent (in this case, the Nao robot), such as the wobbling produced by the walking or in some cases, when the robot missed one mark while moving, the error in the localization was increased in a way that was almost impossible to find the right path to successfully complete the task. This drawbacks lead to a successful rate of only 20% of the rounds.

For the second setup, the robot was assisted by the NRS as described in section 4.3. It was observed that the performance was substantially improved since all the rounds were successfully completed and as a final demonstration of the system, a person who is not familiar with the technical details of this study was taught how to operate and interact with the system.[4] Once again the task was successfully completed. The drawback of this setup is that the system is heavily dependent on the network stability. A failure in the connection or missing information from any of the involved network components may negatively affect the final output.

6 Conclusions

In this work we have successfully integrated a humanoid robot into a NRS to solve a problem in a common day scenario; bringing a can of soda to a human user. This problem involves several tasks such as localization, mobility and

[4] Integrated Project Work demos. See http://www.youtube.com/user/loutfiamy

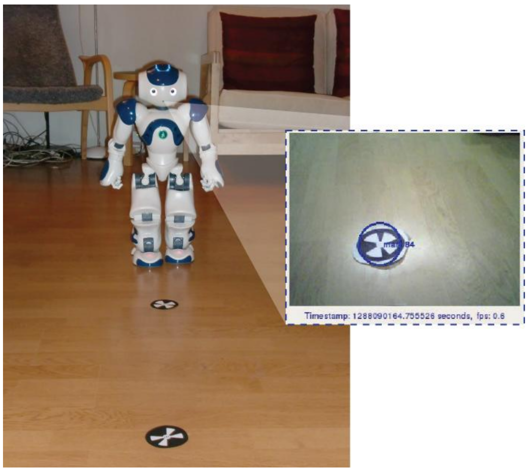

Figure 7: Robot localization using land marks and Land mark detected by Nao*(in sight)*.

cooperative grasping which has to be accomplished in order to successfully achieve the overall goal.

While the Nao is a capable platform; due to the complexity of the problem to be solved and the challenges that home robotics implies, we conclude that the capabilities of the robot can be enhanced and the complexity of the problem can be reduced by decomposing it in simpler tasks executed in cooperation with specialized components in the NRS (i.e. the PEIS Ecology).

The NRS brings also expandability to the Nao itself when new components are added to the network and Nao interacts with them to solve a novel task. The NRS is also beneficiated with the incorporation of Nao, since it became a visible robot that provides different services and acts as an interface between the user and the PEIS Ecology.

Future work may include how to exploit the sensorial capabilities of the Aldebaran's Nao, such as voice and face recognition to allow a more user friendly interface between a smart home and the human user who is requesting different services from the ubiquitous network. This work can also be expanded by using planning and searching techniques to allow the robot to move in non predefined paths and the use of the robot's internal sonars can be used to allow obstacle avoidance.

7 Acknowledgments

This work was supported by AASS (Applied Autonomous Sensory Systems, Örebro University, Sweden) laboratories through the course Integrated Project Work (IPW 2009). Many thanks to all the people at the AASS laboratories for their contributions to this work, especially Dr. Federico Pecora, Dr. Dimitar N. Dimitrov and Todor Stoyanov.

References

[1] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Network robot systems," *Robotics and Autonomous Systems,Elsevier 2008*, 2008.

[2] N. Kubota and K. Nishida, "Cooperative perceptual systems for partner robots based on sensor network," *IJCSNS, International Journal of Computer Science and Network Security*, vol. 6, no. 11, pp. 19–28, 2006.

[3] K. Lynch, I.B.Schwartz, P. Yang, and R. Freeman, "Decentralized environmental modeling by mobile sensor networks," *Robotics, IEEE Transactions on*, vol. 24, pp. 710–724, June 2008.

[4] K. Kemmotsu, T. Tomonaka, S. Shiotani, Y. Koketsu, and M. Iehara, "Recognizing human behaviors with vision sensors in a network robot system," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, (Orlando, FL, United states), pp. 1274 – 1279, 2006.

[5] T. Akimoto and N. Hagita, "Introduction to a network robot system," in *2006 International Symposium on Intelligent Signal Processing and Communications (IEEE Cat. No.06EX1444)*, pp. 91 – 4, 2006.

[6] M. Scopelliti, M. Giuliani, A. D'Amico, and F. Fornara, *I had a robot at home...Peoples' representation of domestic*

robots. In S. Keates, J. Clarkson, P. Langdon and P. Robinson, Designing a more inclusive world. London: Springer Verlag,, 2004.

[7] M. Broxvall, M. Gritti, A. Saffiotti, B. Seo, and Y. Cho, “PEIS ecology: Integrating robots into smart environments,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, (Orlando, FL), pp. 212–218, 2006. Online at http://www.aass.oru.se/˜asaffio/.

[8] A. Saffiotti and M. Broxvall, “PEIS ecologies: Ambient intelligence meets autonomous robotics,” in *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, (Grenoble, France), pp. 275–280, 2005. Online at http://www.aass.oru.se/˜asaffio/.

[9] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Bradford Book, 2004.

[10] J. Davis and V. Sharma, “Fusion-based background-subtraction using contour saliency,” in *CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*, (Washington, DC, USA), pp. 162 –182, IEEE Computer Society, 2005.

[11] R. Gonzalez and R.E.Woods, *Digital Image Processing*. Pearson Prentice Hall, 2008.

[12] Don Murray and James J. Little, “Using Real-Time Stereo Vision for Mobile Robot Navigations,” *Auton. Robot*, vol. 8, no. 2, pp. 161–171, 2000.