Robot Companion for elderly care

Design and Organization of Autonomous Systems
University of Amsterdam, computer science
Supervisor: Frans Groen

Abeer Mahdi
mhammadi@science.uva.nl

Aitor Azcarate Onaindia
aazcarat@science.uva.nl

Ning Yang
nyang@science.uva.nl

Zhiwei Zhan
zzhan@science.uva.nl

Abstract
In the future many elderly people will have to stay longer in their homes. This requires intelligent homes to monitor the elderly, but also need a mobile robot companion, for interaction and adaptive sensing. The possibilities for the use of Aibo as a robot companion are studied in this article. The literature is reviewed for the potential approaches to solve a number of tasks. Some easy tasks are implemented using the script language R-code to study the potential of Aibo.

1. Introduction

In the future many elderly people will have to stay longer in their homes, because the number of places in elderly homes and human care will be limited. To guarantee that they are safe, intelligent equipment is integrated in the house and sensor systems are installed to monitor the elderly.

Intelligent monitoring of the elderly, not only requires fixed systems, but also mobile systems (robots) for adaptive sensing and interaction. In the case of an accident or health emergency the robot can take actions like warning a care center. The robot can also entertain the elder so that the elder doesn't feel lonely.

Further the robot companion should also be able to help a person. It can be done through communicating with the intelligent house or a care center when needed.

This project is concerned with the development of such a mobile personal robot companion, in this case the Sony robot Aibo. We used the Aibo ERS-7 version, which is the latest and most advanced version of Aibo on the current market. The Aibo ERS-7 has 64-bit**,** 576 MHz of CPU, 64 MB of RAM and 4 kinds of input sensors (Head, Back, Chin, Paw). With these features, Aibo has the ability to connect to the home media center: the central computing, internet or entertainment system.

Aibo can be programmed with C++. We can also use Java or matlab; in this case an interface is needed. This interface is called tekkotsu [1], developed by Carnegie Mellon University, US. Another possibility is using R-code, which is a script language, provided by Sony Company. We decided to choose R-code to program Aibo. This is the most suitable tool for us, because it is easy to learn and understand.

There are also other tools such as the motion editor; which is a user-interface that enables the user to make new motions for Aibo. Sony entertainment player is also an existing tool; it is a graphical user interface that controls Aibo via wireless connection. This can for instance be used for remote interaction of Aibo by the nurse.

1.1 Tasks

We made a list of important tasks that we think Aibo must be able to perform, we have divided them in Sensing intelligence, to monitor the elderly, Communication intelligence, to interact with the outside world and Emotional intelligence, that enables Aibo to react to the emotions of the elder;

-Sensing intelligence
- Following a person.
- Detecting when the person has fallen, and then take an action.
- Detect location: Aibo knows where it is in the house and also where it is dangerous to go (for example wet places, walls, and stairs).

-Communication intelligence
- Interaction with persons: using the microphone and the camera of Aibo.
- Interaction with nurse: enable the nurse to monitor the elder via Aibo, and in necessary cases she can talk to the elder and control Aibo.
- Interface with home equipment: Aibo can control the home equipment (like TV) according to the spoken commands given by the elder.

-Emotional intelligence
- Face feature recognition: enable Aibo to recognize the emotions of the elder. In the case that the elder has a sad face, Aibo can dance or play a song to change the elder's mood.
- Initiate a conversation: If Aibo detects the elder is not doing anything, it can begin a

conversation, mention local news or advice the elder to take a walk outside.

- Inactivate mode: if Aibo recognizes the elder is sleeping, then Aibo can recharge or do something else.
- Pro-active suggestions: Aibo can keep an agenda to remind the elder of the appointments. This will be handy for reminding the elder to take medicines.
- Reaction to sound: enable Aibo to react to the commands from the elder with specific actions

We have studied literature on following the elder, localization of Aibo, facial expression recognition and Aibo in the future. To investigate the current possibilities we implemented some simple tasks and gained experiences with Aibo.

The tasks we have implemented are: make Aibo follow the associated pink ball, experiment with this so that it later can be extended by following the elder, pro-active suggestions, make Aibo detect a face and walk to that face, and react to voice commands.

In the next section the results of our literature study will be mentioned and after that we will explain the implementation part and show the results. Finally the conclusion is mentioned.

2. Literature studies

2.1 Following the elder

One of the tasks Aibo has to perform is following the elder and detecting whether the elder has fallen or had an accident. This problem can be seen in an abstract way as the ability to track moving objects. There are a number of algorithms that solve this problem for remote robots.

One of the most popular techniques to track moving objects is mean-shift analysis. Mean-shift is a nonparametric estimator of density gradient, which is employed in the joint, spatial-range domain of gray level and color images for discontinuity preserving filtering and image segmentation [2]. The tracking of a non-rigid object is based on visual features such as color and/or texture, whose statistical distributions characterize the object of interest. The mean shift iterations are employed to derive the object candidate that is most similar to a given model while predicting the next object location. This method provides accurate localization, and it is computationally fast.

However, the mean-shift tracker is not automatic since it requires initial model properties, i.e. object boundary, etc. There is also a problem when the robot loses the sight of the object, for example when the object turns suddenly to a side. The Robot then cannot know which side it must look at to find the object. This also happens when the background color is the same as the color of the object, when the object then moves; the robot cannot detect this movement.

Another solution is to use the optical flow algorithm which consists in extracting a dense velocity field from an image sequence assuming that intensity is conserved during displacement [3].

The problem with this algorithm is when there are many motions occurring at the same time. The robot doesn't know which object it must track. It also requires the robot to stay still, because any movement of the head or the body can affect the factor field. This can however be filtered using other algorithms.

A more trivial algorithm is to follow the direction of the voice of the elder that is been followed. The robot can be programmed to recognize its owner's voice. By using microphones and a direction of hearing detecting algorithm, the robot can easily follow the elder. One drawback of this algorithm, in our opinion, is its sensitivity to noise. It also means that the elder has to speak all the time, so that the robot can be able to follow him/her. This algorithm will be useful when integrated or combined with one of the algorithms mentions above.

The most suitable way in our point of view is to implement a hybrid of the two algorithms. The mean-shift algorithm is then used to track the object in normal conditions, when the robot loses sight of the object it can then apply flow motion algorithm on the previous images. Unfortunately there are not many studies done in this area, where we can come to know if it is good or not.

In our case, the simplest solution is to track a color histogram. The method dependent upon lighting conditions and requires characteristic color of the shirt or cap, which the elder must always wear.

2.2Localization of Aibo

In this part different methods for detecting the location of the Aibo inside the house will be discussed. Two different kinds of methods are the map based method and the appearance-based method.

The map based method uses an internal map of the house (or building) to localize the Aibo. The detection can be done first by using sensors and identifying packets from a beacon [4,5,6,7] and second by using a chip [8] inside Aibo that can broadcast unique identification codes.

The first method employs beacons to broadcast identifying packets from strategic positions within a building infrastructure in such a way that each resolvable position is covered by a unique collection of beacons; a user of such a system can thus determine his location by means of the beacon packets received.

With the second method a chip is built in Aibo. When queried by a reader device the chip broadcasts a unique identification code. At present the chips can return such a signal from distances up to a few tens of meters depending on the communicating frequencies and transmitting powers involved. The tags may be as small as rice grains.

In the appearance-based method [9,10] the position of the Aibo is determined directly comparing the last observed image with images observed before. The position at which the previous observed image was taken is known. The appearance-based localization departs from a training set including images shot at known positions. The dimensionality of the images on this training set is reduced using a PCA [11] process for efficient storage. When the robot is moving in the environment, the collected images (after the corresponding dimensionality reduction) are compared with those in the training set and a particle filter algorithm is used to determine the position of the robot.

The different methods have advantages and disadvantages. The appearance-based method doesn't need a predefined map of the building but it will need a lot of sensor values (in this case pictures) in advance. It will thus need a period for learning how the building is constructed and it will also need lots of pictures taken in advance of the building it is going to work in. The appearance-based methods do a fairly well job in static environments. In real life (where the robots eventually will work in) the environments aren't static so some work still needs to be done in the non-static ('real life') environments for appearance-based methods.

In the map based method an extensive modeling of the environment is needed. This is only feasible if sufficient prior knowledge (where tables, chairs etc. will be) about the environment is assumed. The method thus depends strongly on the environment the Aibo will be moving in.

These methods can be combined [12]. This method will then use the previously stored information to reduce the uncertainty on its position. The improvements on the position can be back propagated to map points in previous time slices. This will result in a correction on Aibo's position and the map. This will thus achieve the objective of concurrently localize and build a correct map of the environment.

The on-line construction and update of the map allows overcoming the major hurdles of traditional appearance-based localization. First, the robot can operate in previously unknown areas. Second, we can deal with changes in the environment: new observations obtained at already explored positions are added to the map, the old observations at those positions are not used any more and they are slowly forgotten. Finally, the way in which the map is built guarantees a uniform sampling of the feature space and not of the geometric space, as it happens in normal appearance-based localization.

The map based methods have some disadvantages. Every time something is changed in a room, the map will also have to be changed. The map based method will need sensors to localize but because Aibo will be in an 'intelligent home' these sensors will already be available.

The appearance based method doesn't need a predefined map of the building it will work in to localize itself. Thus with this method it is possible to localize in different buildings. This will be the case for Aibo because every home is different. Although this advantage, the appearance based methods still don't work well in non-static environments. Thus for localization a map based method is preferred.

2.3Facial expression recognition

Being a robot dog, recognizing the owner's face is an important feature as it is essential to the robot dog to interact differently on different people. In addition to this, recognizing owner's facial expression and voice emotion would be nice if possible. With these features Aibo would look like a real dog in terms of feeling and emotion.

Face detection and expression recognition need to be used to interact with human's facial expressions. Face to face communication of robot with human is a real-time process operating at a time scale of 40 milliseconds. This makes it necessary for robots to rely on sensory rich perceptual primitives. At the University of California, a system has been developed that can automatically detects faces in the video image in real time and then distinguish them into 7 categories [13]: neutral, anger, disgust, fear, joy, sadness, surprise. The system consists of a face finder and an expression recognizer. The system has been deployed on some platforms including Aibo.

The system has a real-time face-detection system developed based on a visual object detection framework that is capable of processing images rapidly while achieving high detection rate [14]. This base system operates on 384 by 288 pixel images. Faces are detected at 15 frames per second on a 700MHz Pentium III, making the system match the technical specification of Aibo well.

After locating faces, it requires preprocessing to convert images into a standard pattern presentation to be processed later by facial expression classification. Here a few internal facial features are used, making the overall system very fast.

Facial expression classification was based on support vector machines (SVM's) [15]. SVM is a popular technique for classification. It is considered easier to use than neural networks. A classification task usually involves training and testing data which consist of some data instances. Each instance in the training set contains one "target value" (class labels) and several "attributes" (features). The goal of SVM is to produce a model which predicts the target values of data instances in the testing set which are given only the attributes. In the facial expression classification stage, a SVM was trained to discriminate between emotions. Then the emotion category decision can be performed to categories emotion into 7 kinds.

The system also has a real time emotion mirror that renders a 3D character that mimics the facial expression of people.

According to [13], real time facial expression recognition is an achievable goal with Aibo's computing power. The system has been deployed and tested on Sony's Aibo and other robots. At the moment, the performance is being evaluated.

2.4 Aibo in the future

In the future, with the advances of sensor and artificial intelligence technologies, Aibo can easily recognize different users as well as their surrounding environment. And it can think on its own and react to (or, possibly even predict) different user's individual needs. Next we discuss a scenario in the future where Aibo serves as a universal interface to all electronic home equipment.

Future homes will be equipped with loads of next generation wireless-enabled electronic devices such as TV, MP3/DVD player, digital recorder, radio, telephone, refrigerator, climate control, alarm, lights, etc. Each of them can be configured and customized to serve individual needs of different users. However, different vendors may adopt different standards of controlling and operating their devices. And it is a great trouble for the users to configure and operate each device. The problem can be solved by a future generation of robots. It will serve as the universal control interface between the users and all these equipments.

Here is one possible scenario of the interaction between Aibo and the home equipment in the near future: Tom gets home from work. Aibo recognizes him and greets him at the doorway and disables the invasion alarm system. Tom sits down in the couch and wants to listen to some music. Aibo controls the MP3 player and plays Tom's favorite albums. After a short while, Tom's wife Amy and their 5 year old Stephanie get home. Amy wants to have a bath so that Aibo automatically adjusts the temperature in the bathroom and runs the bath to her favorite temperature. Stephanie wants her daddy to watch Mickey Mouse with her and Aibo will turn off Tom's music and asks the TV and DVD player to start playing. Kitchen counter has an over-head LCD and Amy's dinner recipe for tonight has been displayed. Tom knows every ingredient is already there in the refrigerator because otherwise Aibo would have already notified him via his cell to pick stuff up on his way back home. When the family is enjoying the dinner, Tom wants the telephone to be silenced and all the incoming calls will be directed to their answer machine. Numerous things like above can be listed. The robot companion will certainly make people's daily life better in the future.

Aibo's ability of controlling and operating different home equipment is based on the technological advances of many fields. Let's take the example where Tom gets home and Aibo recognizes him and plays his favorite song for him as an example to illustrate what technologies are involved and what needs to be done.

- Face recognition: Nowadays Aibo is able to do face detection. However, in the future, face recognition is a much needed ability if this robot pet is to be put into a broader range of application scenarios. This requires that Aibo is backed by a non-trivial image database, which stores pictures of all family members of Tom, and their friends. When a face is detected by Aibo, it should be matched against the stored pictures in the database in order to decide who exactly is facing Aibo at the moment. Interesting face recognition techniques can be found in section 2.3, [18], [19] and many other related publications.
- Distributed Computation: Aibo's size and power limits its ability of running complex computation and expensive algorithms. Therefore, all major computations are done in the backend server. This including the face recognition algorithms and the image database. The presence of a back-end server also supports the possibility of multi-Aibo environments where more than one robot pet dog can collaborate together to finish a task in the future.
- Multimedia Database: After Aibo's successful recognition of Tom's face, it should be able to know what Tom's favorite songs are. A multimedia database is required in order to fulfill the needs. The database is running on the backend server and consists of all the songs Tom's family has. It also organizes the songs into categories based on user's grading or access counts where the concept of "Tom's favorite" can be easily interpreted.
- XML and Standards: In the future, the communication between Aibo and all the devices should follow a standard and they should all speak the same universal language, for example XML. Therefore, the whole Aibo-centric future home can be extensible and all kinds of devices can be easily integrated into this system, as long as they support XML.
- Fault-Tolerant System: Each of Aibo's tasks can fail due to unforeseeable reasons. One common example can be that the wireless connection between Aibo and MP3 player is temporarily unavailable. Aibo should be able to report this failure to the user. Other failures like backend server crash can result in catastrophic mess, i.e. digital media loss; face recognition engine not functioning, etc. Therefore, the storage system of the backend server should be persistent and fault-tolerant. Replicated server architecture may be feasible in this scenario. More information on fault-tolerance and replication can be found in [20]. It is not the essential part of the application.
- Security: Home is a highly private area. Aibo's wireless communication with other devices and the backend server should be encrypted to prevent

eavesdropping. The authentication and authorization to the system should also be enhanced by mature techniques such as fingerprint scan [21] and/or passwords.

Nowadays , this scenario only exist in the laboratory of some research institutions or universities, like the “Homelab” of the Philips Company [16] or “The Aware home” of the Georgia institute of Technology , US [17]. A lot of effort is still needed to make this kind of intelligent home come into people’s daily life. Because this needs not only the techniques from the field of artificial intelligence, but also the knowledge from other aspects, like building architecture, etc.

3. Implementation

Our implementations were done in R-CODE SDK. R-code has a number of advantages; it is a simplified scripting language, which is easy to learn in a short period. It also can be debugged at run-time via wireless LAN. On the other hand, R-CODE is not suitable for complex scientific calculations since current available functions are limited. It is also not open-source. Predefined functions can not be altered and adding new functions is also not allowed.

In the following we will explain the implementation of tasks, and also mention the results of our experiments.

3.1Following the ball

One of the tasks that we want Aibo to do is following the elder. It is useful that Aibo stays in the same area as the elder, for example to hear the voice commands, face detection, etc.

This problem can be realized by color histogram tracking. It can be simplified by following a pink ball and extending that to make Aibo follow the elder. Perhaps by putting a pink marker on the elder’s shoes or back.

In R-CODE, some functions related to searching and tracking are predefined. We used the functions in the following way:

```
// tracking the pink ball
start searching
   if pink ball located
        get the position of the ball(angle and distance)
        turn body(angle)
        walk to the ball(distance)
   else search again
```

Results

We have experimented with the code and we noticed that this code is not accurate. During the experiment we have noticed that the WALK – function already implemented in R-code is not accurate. When running the function, Aibo does not walk in a straight line (figure 1).

The deviation of Aibo is about 10 degrees. If Aibo has to walk a large distance (about 10 meters) the deviation will have a large effect on the end position. Thus if it wants to get to a certain position it will have to re-measure the distance a couple of times to get closer to the desired end position.

Aibo loses the pink ball mostly due to the deviation in its walking. The distance to the pink ball isn’t measured that correctly either. It will often get closer to the ball than we have programmed it. The bigger the distance to the pink ball is, the bigger the error is. We programmed it to stay at a distance of 10 cm but often it will walk nearer to the ball.

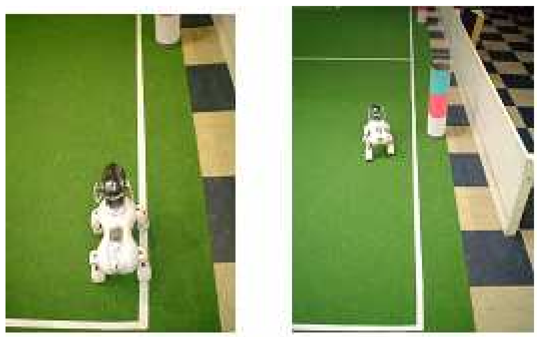

Figure1 Aibo walking, On the left side is the image of Aibo at the starting position, and on the right side is the final position After executing the walk function

3.2 Pro-active suggestions

Pro-active suggestions are necessary to remind the elder of appointments or tasks the elder must do, such as taking medicine. Aibo is then a digital Agenda for the elder.

R-code has a clock-function. It updates an integer every 32ms. R-code also has a second, minute, hour, day, week, month and year variable, that give the exact time when used. We have experimented with both in our implementation.

Results
We have noticed that working with the variables is better, thus we made Aibo give a message to the elder at a given time and date. It will give messages accurately according to the time. However, when using the clock function, this is not so simple because the time must be calculated to milliseconds.

3.3 Reaction to sound
Reaction to sound is obviously an important technique for Aibo. By using this approach, Aibo can listen to and recognize the voice commands the elder gives. In R-code system, there is a predefined "Voice ID List" which lists all the voice commands that can be recognized by Aibo.

We programmed Aibo to perform an action or play a sound when it recognizes the specific voice commands. We recorded our own voice by using the "Sound Recorder" provided by Windows XP and converted these .wav files to "8k 8bit MONO PCM" format, which is one of the formats that can be played.

We also have tried to use the latest version of Sony's Motion Editor to create new motions for Aibo. But we noticed most of the motions don't work well in Aibo, although they did work correctly in the simulator of the Motion Editor. We think it's because Motion Editor doesn't support Aibo ERS-7 very well.

The structure of our program is:

Do
Switch voice id:
Case “Aibo”: Play “I’m here”
 ……….
Case “That’s wrong”: Play “Sorry”
Case :else: Play ”repeat please”
Loop

Below is a sample list of some commands we made Aibo react to:

Voice List:	**Answers /Motions we created**
Aibo	I'm here
Hello, Say hello	Hello
What's your name?	Aibo
Turn left	Aibo will turn it’s body to left
Turn right	Aibo will turn it’s body to right
Commands not recognized	Repeat please

Results
Due to the limits of R-code, Aibo can only recognize the spoken commands from the voice id list of R-code system. We think Aibo should be able to recognize more voices if it will be really used in the field of elderly care. In our experiments, if the voice commands aren’t spoken clearly, Aibo will often play sounds or actions it doesn’t have to play.

Through our experiments, we noticed that voices are recognized accurately within a distance of about one meter. But if there is some background noise, voices will not be recognized or recognized wrongly. In such case, repeating the command will make Aibo work correctly most of the time.

3.4 Face Detection
Face detection is a kind of emotional intelligence of Aibo, and gives the elder the feeling that Aibo is not just a robot. Face detection is also very useful when an elder falls down. In this case actions should be taken after a human face was detected on the floor (i.e.: somebody falls down). It is easier to implement this task using R-code because there is a system variable “Face” in R-code. It indicates whether a human face is detected or not. We have implemented this task and made Aibo detect a face and walk towards it. The algorithm is:

```
// searching for a face
 start searching
   if face located
       get the distance to the face
       subtract 100 from distance
       walk to the face(distance)
```

else search again

Results
In our experiments, the program worked most of the time when there is a human face in front of Aibo. But we also noticed that sometimes, Aibo acts like it detects a face while actually there is no face in front of it. This happened a few times when we did the experiments. We think that the reason can be the predefined face detection function in R-code is not accurate enough.

4. Discussion

In the experiments we did with the Aibo we noticed a few drawbacks of the R-code. When using the predefined walk function, Aibo has a small deviation to the side. We also noticed that the face detection function is not very accurate.

The implementation language we used (R-CODE) still is too basic to really program Aibo to perform intelligent tasks. Aibo can be programmed in more powerful languages. Due to time and the simplicity of R-CODE to show some basic actions we didn't use the other languages(c++, java or matlab). To give a feeling of the complexity of the other languages; we found a file written in C++ that makes Aibo move it's head, the file was 400 lines of code.

Aibo isn't ready to be used in elderly care with the software delivered by Sony. It includes an entertainment player that can control Aibo via wireless LAN. With this program it could be possible for a nurse to control Aibo when it detects an accident. In order to use this, an interface between the computer at the home of the elder and the computer at the nurse's office would have to be developed. The developers will have to keep in mind security problems etc. which are common in programs that give access to a computer via internet.

For the pro-active suggestions it could be better to use the pc as an agenda and transmit an appointment or task via the wireless LAN. This could also use the entertainment player. Here also someone will have to develop the interface between a scheduler (such as outlook) and the entertainment player provided by Sony.

The software delivered with Aibo is still too basic to use Aibo in elderly care. With adjustments made on the software we think that Aibo will be usable in elderly companion.

5. References

[1] Tekkotsu Homepage: http://www-2.cs.cmu.edu/~tekkotsu/

[2]D.Comaniciu and P.Meer; *Mean Shift Analysis and Applications*, http://www.caip.rutgers.edu/riul/research/papers/pdf/spatmsft.pdf

[3] G. Quenot ,*Computation of optical flow*,http://www.limsi.fr/Individu/quenot/publis/publis.html/

[4] Launch of Hitachi AirLocation(TM), *a Wireless LAN-Based Location Detection System*
Achieving Highly Accurate Positioning with Small Error Range of 1 to 3 Meters
http://www2.hitachi-cable.co.jp/apps/hnews.nsf/0/919cbdc6e71ce71a49256e200083b968?OpenDocument

[5] R. Ungrangsi, A. Trachtenberg, and D. Starobinski
An Implementation of Indoor Location Detection Systems based on Identifying Codes,
http://ipsit.bu.edu/documents/web_location.pdf

[6] Chris Malladi, Rachanee Ungrangsi, Professor Ari Trachtenberg
Robust Location Detection using Identifying Codes
http://ipsit.bu.edu/nislab/projects/locdetect/index.htm

[7]Saikat Ray, Rachanee Ungrangsi, Francesco De Pellegrini, Ari Trachtenberg, David Starobinski,
Robust Location Detection in Emergency Sensor Networks
http://www.ieee-infocom.org/2003/papers/26_01.PDF

[8] Matthew James , *Where are you now? Location detection systems and personal privacy,*
http://www.aph.gov.au/library/pubs/RN/2003-04/04rn60.htm

[9] Ben Krose, BasTerwijn and Josep M. Porta, *Autonomous Robot Localization Using Vision,*

IAS Group,Informatics Institute, University of Amsterdam, 1098SJ, Amsterdam, The Netherlands
http://staff.science.uva.nl/~porta/ambience/localization/index_en.html
[10] B.J.A. Kröse, R. Bunschoten, N. Vlassis, and Y. Motomura**.** *Appearance based robot localization*. In G Kraetzschmar, editor, *Proc. IJCAI-99 Workshop on adaptive spatial representations of dynamic environments*, pages 53-58, 1999.
[11]Lindsay I Smith
A tutorial on Principal Components Analysis
http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
[12] Josep M. Porta, Ben J.A. Kr¨ose, *Appearance-based Concurrent Map Building and Localization using a Multi-Hypotheses Tracker* , IAS Group, Informatics Institute, University of Amsterdam, 1098SJ, Amsterdam, The Netherlands
http://www-iri.upc.es/people/porta/Docs/iros04.pdf
[13] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel and Javier R. Movellan. *Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction.* Machine Perception Laboratory, Institute for Neural Computation & Intelligent Robotics and Communication Laboratories_ ATR, Kyoto, Japan.
[14] Paul Viola and Michael Jones. *Robust real-time object detection*.Technical Report CRL 2001/01, Cambridge Research-Laboratory, 2001.
[15] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin ,*A Practical Guide to Support Vector Classification*, Department of Computer Science and Information Engineering, National Taiwan University
[16].Philips Research's HomeLab Project: http://www.philips.com/homelab
[17]. Georgia Tech's AwareHome Project: http://www.cc.gatech.edu/fce/ahri/
[18]. Wen-Yi Zhao, Rama Chellappa, P.J. Jonathon Phillips, and Azriel Rosenfeld , *Face Recognition: A Literature Survey*, ACM Computing Survey, December Issue, pp. 399-458, 2003.
[19]. MingHsuan Yang and Narendra Ahuja, *Face Detection and Gesture Recognition for Human-Computer Interaction*, Kluwer Academic Publishers, 2001.
[20]. R. Guerraoui and A. Schiper. *Fault-Tolerance by Replication in Distributed Systems*. In Proc Conference on Reliable Software Technologies (invited paper), pages 38--57. Springer Verlag, LNCS 1088, June 1996.
[21].International Biomatric Group's Fingerprint Technology: http://www.biometricgroup.com/reports/public/reports_finger-scan.html