Visual Perception System for a Social Robot (VISOR)

Deliverable 4

FINAL REPORT

Luis Molina[1], Juan P. Bandera[1], Pedro Núñez[1], Jörg Rett[2], Antonio Bandera[1], Francisco Sandoval[1]

[1]Grupo de Ingeniería de Sistemas Integrados

Dpto. Tecnología Electrónica, Universidad de Málaga

Campus de Teatinos 29071 Málaga (Spain)

www.grupoisis.uma.es

[2]Institute of Systems and Robotics

University of Coimbra

Polo II, 3030-290 Coimbra (Portugal)

www.isr.uc.pt

Due date of deliverable: September 15th, 2006
Actual submission date: September 30th, 2006
Start date of project: September 15th, 2005
Duration: 12 months

Organizational name of responsible for this deliverable:
Grupo de Ingeniería de Sistemas Integrados
Revision: 1.2
Dissemination level: PU

Contents

List of Figures

1 Executive summary

The European Research Network - EURON - launched the Topical Research Study "VISOR - Visual perception system for a SOcial Robot" in June 2005, under its second call for proposals. The project was officially launched by EURON on September 15th, with a duration of one year.

Two partners were involved in the project: Ingeniería de Sistemas Integrados, of the Departamento de Tecnología Electrónica, at the University of Málaga (Spain); and the Instituto de Sistemas e Robótica of the Departamento de Engenharia Electrotcnica at the Universidade de Coimbra (Portugal).

The aim of this research study was to investigate visual, human-oriented perception skills for a social robot. In particular, this project investigated how the requirements for accomplishing a visual task determine the optimal architecture of a vision system. The issues that were considered were human and object representation, and the selection of low-level and high-level vision features that are required for such representations. Relevant technical achievements in those areas have been produced (see Section 4), which have been disseminated in journal and conference papers, and have influenced a number of Ph.D. projects (See Section 7). This project has partially funded researcher exchanges (See Section 5) which have resulted in an ensemble effort to implement two vision architectures which integrate the subsystems investigated (Section 4).

There have been two trial sites where a series of experiments demonstrating the developed vision architectures where shown to the public, at University of Mlaga and University of Coimbra. (See Section 5.2).

2 Project objectives

The visual perception system of a social robot is the responsible of solving several complex tasks such as the human faces identification, head and hands motion capture, gesture recognition or the reading of facial expressions to emulate human social perception. This information permits that the robot be able to identify who the human is, what the human is doing, how the human is doing it and even to imitate the human motion. Besides, these human-related tasks must be run in parallel with object-related ones, which permit the robot to recognize objects extracted from the environment. This

supposes a high computational load which must be efficiently managed in order to achieve a fast, natural response of the robot in the human-robot interaction cycle.

Computer vision research has traditionally emphasized on investigating each module of the whole visual perception system as a general and isolated item. Such research efforts often generate unrealistic solutions from ill-defined assumptions. In contrast, this research study has considered the visual perception system as a whole and its main objective has been the development of a task-oriented vision system. This system has finally oriented its resources to solve two different problems: a human-robot interaction scenario, where the robot must interact with humans which will guide the robot's actions using gestures, and a human upper-body motion capture framework. In both cases, our research has been focused on interactions among modules as well on each individual vision module. In order to achieve these goals, this research study has accomplished the following tasks:

- *Face recognition.*

  Face detection and recognition are two essential tasks which must be assumed by the visual perception system of a social robot. Although other popular approaches propose to look for faces in the whole input image, our system has reduced this search to a set of regions of the image. These regions present a high density of skin-coloured pixels. Then, the skin colour detection, face detection and face recognition modules can be located in the three different levels of our vision architecture and they constitute a reduced version of the whole system. The skin colour detection module is a low-level module, which must be integrated in the pre-attentive stage of the vision system. The face detection looks for faces in certain regions. It is also a low-level module, but it must be positioned in a higher level: the semi-attentive stage. Finally, face recognition can be considered as a high-level module. It is located at the attentive stage of the vision system.

- *Attention mechanism.*

  As it is shown in the anterior item, the data acquisition and the information extraction processes are closely related in the proposed vision system and they depend on the current task. Following this assumption, the perception process becomes an active mechanism that extracts the most relevant information from the huge amount of input

data depending on the final application. This selection or pre-attention mechanism allows to efficiently exploiting the available computational resources either by dedicating all of them to a specific perceptual task or by sharing them among a small set of tasks. In this research study, a general purpose attention mechanism based on the feature integration theory [1] has been developed. This mechanism is capable of handling dynamic environments, and detecting both human faces and hands and objects of interest in a fast way. It divides the whole vision system in the three levels previously mentioned: pre-attentive, semi-attentive and attentive stages. Each stage could be composed by several modules.

- *Hierarchical tracking of head and hands: Gesture recognition.*

  Other very important task that must solve the visual perception system is the ability to capture the human motion. It is especially interesting to track the human hands because, in addition to facial expressions, non-verbal communication is often conveyed through gestures and body movement. Human hands are non-rigid objects with many degrees of freedom and can, through different postures and motions, be used to express information. In order to achieve a fast tracking of non-rigid objects like human's head or hands, a pyramid structure has been modified to increase its efficiency: the Bounded Irregular Pyramid (BIP) [2]. The BIP is a mixture of regular and irregular pyramids whose goal is to combine their advantages: low computational cost and accurate results. Thus, its data structure combines a regular decimation process with an union-find strategy to build the successive levels of the structure. The irregular part of the BIP allows to solve the three main problems of regular structures: non-connectivity preserving, non-adaptability to the image layout and shift-variance. On the other hand, the BIP is computationally efficient because its regular part prevents a big increase of height.

- *Human motion capture system.*

  The human motion capture problem has been accurately solved using marker-based systems that usually require the human to wear especial gear and move within a constrained capture space. However, in this research study, the detection and tracking of the demonstrator upper-body movements has been solved using a marker-less approach which assumes that is not necessary to know with precision the movements

of all joints to achieve an accurate human motion capture. Although some precursors of this system [3][4] assumed that it was enough to track the movements of the head and the hands to obtain the global pose of a human upper-body, experimental results have finally shown that more information regarding elbow pose and body orientation is needed, in order to provide a better estimation of human movements. Thus, the proposed architecture still considers the 3D movements of the human head and hands as the most important features to recover his motion, but it also processes silhouette information to provide approximated positions for the elbows and an overall body orientation. Finally, it has been necessary to include a model-based pose estimation module to remove inconsistent data. This pose estimation method uses the information provided by the vision module to compute a set of joint angles. These angles are obtained using a constrained inverse kinematics algorithm. The analytic nature of this method allows it to offer the required joint angles on real-time.

- *Robotic head control module.*

  The hardware and software of a robotic head with three degrees of freedom (pan, tilt and vergence) has been developed. Using this robotic head, the visual system could interact deliberately with the environment by controlling the gaze and moving the focus of attention. However, it must be pointed out that this head has not been finally employed. In January 2006, the project provided us a stereo vision system from Videre design. This system includes two cameras in a fixed position and it permits to obtain depth information at 30 images per second. This static head imposes us the impossibility of tracking any item which goes out the field of view of the vision system.

- *Task-oriented control architecture.*

  Since visual sensing is performed with limited resources, visual strategies must to be planned so that only necessary information will be obtained. The generation of the appropriate visual strategy entails knowing what information to extract, where to get it, and how to get it. This assumption is the base of the proposed architecture: the pre-attentive stage of the attention mechanism extracts the most relevant information from the huge amount of input data (e.g. skin colour regions), the semi-attentive stage makes a first classification of the regions

of interest and it permits to track the movement of these regions, and the attentive stage uses this data to achieve more complex behaviours, like face or gesture recognition and upper-body motion capture.

3 Relationship with Euron objectives and other projects in and out of Euron

The objective of the Euron is "to ensure that adequate resources and mechanisms are available to enable Europe to become the leading area in robotics". In this sense, VISOR has contributed with the following items:

- **Research coordination** at all levels between two European robotics research labs, the Ingeniería de Sistemas Integrados, of the Departamento de Tecnología Electrónica, at the University of Málaga (Spain); and the Instituto de Sistemas e Robótica of the Departamento de Engenharia Electrotcnica at the Universidade de Coimbra (Portugal).
- **Education and trainining**, by partially funding PhD programmes at both research labs in Coimbra and Mlaga, and funding researcher exchanges between them.
- **Dissemination**, through public deliverables and papers at international conferences and refereed journals. VISOR has also organised a workshop on Visual based Human-Robot Interaction, which was held in conjunction with EUROS 2006.

Euron Call 1 for proposals funded the Prospective Research Project also funded PHRIDOM, which looked at Physical Human Robot-Interaction from the point of view of safety in cooperation between robot and human. In contrast, VISOR has focused on visual human-robot interaction for social robotics.

Outside of Euron, this project is highly related to the first stage of the Project n. TIN2005-01349 from the Spanish Ministerio de Educación y Ciencia (MEC), which the ISIS group in Mlaga is currently unfolding. This project proposes the development of a control architecture for humanoid robots aimed at the automatic learning of sensory-motor skills. This architecture will be organised as a horizontal hierarchy of control layers where each control layer will implement a sensory-motor primitive designed to solve a

specific task. In this system, an active vision system will be developed to filter the amount of visual information processed to estimate the state of from complex, dynamic environments. Learning will be implemented as a special primitive whose responsibility is the generation of new primitives by imitation. The goals of this primitive will be: i) capturing and imitating the motion of a teacher performing a particular task and ii) estimating the changes to the environment that the teacher has caused, to a point where the particular task performed by the teacher can be reproduced by the robot. Therefore, the visual perception system must be capable of solve tasks like the human motion capture or face recognition.

Finally, this research project is aligned with the objectives of the programs Beyond Robotics Proactive Initiative (COGNIRON project), e-inclusion and Technology-enhanced Learning, inside the priority IST (Information Society Technologies of the VI Frame Program of the European Union).

4 Technical achievements of the project

4.1 Introduction

A social robot can be defined as an embodied agent that is part of a heterogeneous society of robots or humans. Therefore, a social robot must be able to recognize humans or other social robots and engage in social interactions. To achieve this interaction, social learning and imitation, gesture and natural language communication, emotion and recognition of interaction partners are all fundamental factors. This implies that the vision system of this social robot must be capable of solving the problems of identifying faces, measuring head and hands poses, capturing human motion, recognizing gestures and reading facial expressions. There are two different ways to accomplish the development of the visual perception system of a social robot: the reconstructive vision paradigm and the animate vision paradigm.

In the reconstructive paradigm, visual perception must recover information from the whole scene. Thus, vision typically consists of three consecutive actions: reconstruction of physical, real scene parameters from the image input, segmentation of the image into regions, and description of this input. Then, general tasks can be accomplished using higher-level modules which act on this provided description. This process requires a huge amount of computational capacity to manage the visual data acquisition and process-

ing. On the other hand, animate vision proposes to dedicate all computational resources to solve a small set of specific perceptual tasks. That is, in the animate paradigm, visual perception becomes a task-oriented process: if visual sensing is performed with limited resources, they must be focused on extracting only necessary visual information. To achieve this, the responses to what information to extract, where to get it and how to get it, must be known in advance. Besides, animate vision proposes that vision must operate continuously and it must furnish results within a fixed delay. Rather than obtain a maximum of information from any one image, as it proposes by the reconstructive paradigm, the camera is an active sensor giving signals that provide only limited information about the scene.

In this research study, visual perception is achieved following the basics of animate vision. Thus, the information extracted from the input image depends on current tasks. That is, the perception process becomes an active mechanism that extracts the most relevant information from the huge amount of input data depending on the application. If social behaviour imposes a predefined set of visual-based tasks, the selection mechanism allows to efficiently exploit the available computational resources by dedicating all of them to the corresponding small set of tasks. Two examples of complex visual-based social behaviours have been accomplished in this research study: i) a human motion capture application which would permit to the social robot the imitation of human activities, and ii) a human-robot interaction scenario, where a set of humans controls by manual gestures the activity of a robot. Both behaviours lie over task-oriented visual architectures which will be described in next sections.

4.2 Overview

Each of the scenarios defined (human motion capture and human-robot interaction) has led to an architecture which specializes in that application.

4.2.1 An attentional architecture for visual human motion capture

Fig. 1 depicts an overview of the proposed architecture. The whole architecture can be divided into three major modules related to the pre-attentive, semi-attentive and attentive stages. Thus, the visual perception stage develops a general purpose attention mechanism based on the feature integration

theory, which is capable of handling dynamic environments. The first and second stages detect and track human faces and hands in a fast way. The third stage recognizes faces and gestures and performs human upper-body motion capture. The pre-attentive stage determines and selects salient image regions by computing a number of different features. The semi-atttentive stage makes use of object specific properties to filter out data and only track significant objects. Thus, the system is related to the Backer and Mertsching's proposal [5] in several aspects. The first is the use of a pre-attentive stage in which parallel features are computed and integrated into a saliency map. However, in contrast with this and other attention systems, we have introduced the skin colour as input feature in order to detect human faces or hands as possible regions of interest. Thus, in this work, skin colour is first detected using a chrominance distribution model [6] and then integrated as input feature in a saliency map. Other similarity is that this pre-attentive stage is followed by a semi-attentive stage where a tracking process is performed. But, while Backer and Mertsching's approach performs the tracking over the saliency map by using dynamics neural fields, our method tracks the most salient regions over the input image with a hierarchical approach based on the Bounded Irregular Pyramid [2]. The output regions of the tracking algorithm are used to implement the inhibition of return and avoid revisit or ignore objects. The main disadvantage of using dynamic neural fields for controlling behavior is the high computational cost for simulating the field dynamics by numerical methods. The Bounded Irregular Pyramid approach allows real time tracking of a non-rigid object without a previous learning of different objects views [2]. Besides, the tracking approach can work simultaneously with several regions without a high increment of the computational cost. Finally, the attentive stage can recognize faces and gestures and it also detects and tracks the demonstrator upper-body movements.

4.2.2 *Nicole*: an architecture for social interaction

Nicole is a multipurpose platform to investigate social interaction between humans and robots. For the VISOR project, we have defined three goals:

First, to show the development of a system to recognize gestures from a stream of camera images using a Bayesian framework. Second, to create a scenario where interaction takes place between several people that play with Nicole based on mapping of those gestures to actions performed by Nicole (Nicole@Play see fig. 2). Third, to create a scenario where the face of a per-

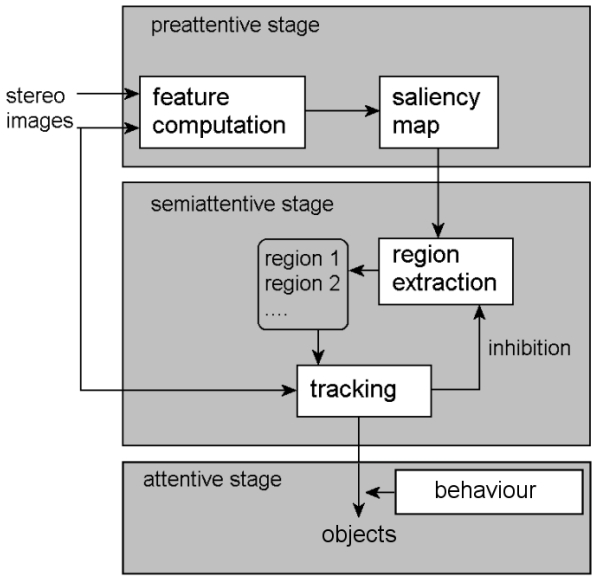


Figure 1: Overview of the proposed vision system

son can be learned automatically by an intelligent environment where Nicole can interact with a specific set of people known as godfathers (Nicole@Face, see fig. 3). Other scenarios are thinkable like the already well explored guide robot context [7], [8], [9].

We present a system that extracts the gesture-features of a human actor from a series of images taken by a single camera. Figure 4 shows the architecture of our system with the camera placed inside the 'Observation Level'. Depending on the scenario (Nicole@Face) we also incorporate face recognition into the system. The hands and the face of the actor are detected and tracked automatically without using a special device (markers). The system is based on implementations to capture human motions of hands and head using the open-source library OpenCV from Intel. From the motion trajectories we extract features like displacements (in pixels) of hands and head. The 'Recognition level' is based on a Bayesian method to find the most likely gesture that might have created the observed sequence of features using the commercial ProBT library from ProBayes. The script for the scenario is placed inside the 'Action Planner Level' which is in ultimate control of the process.

The rest of this section summarizes the main aspects of the components of

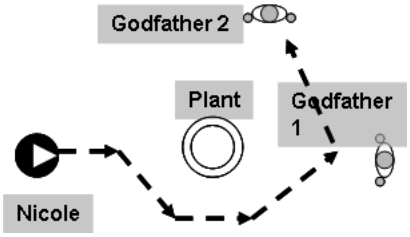


Figure 2: Scenario 1: People play with Nicole.

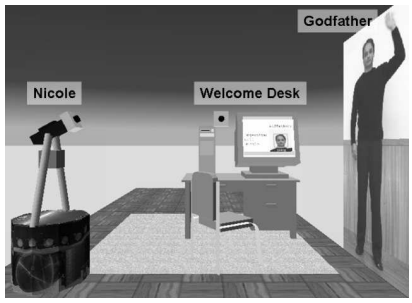


Figure 3: Scenario 2: Nicole can recognize a godfather.

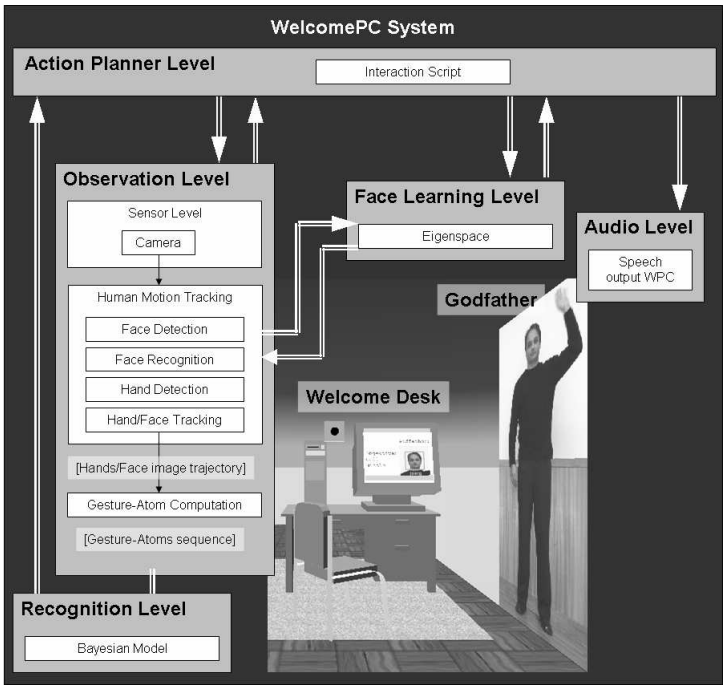


Figure 4: System architecture of the Welcome Desk

both architectures. For further details please see the previous VISOR public deliverables.

4.3 Skin colour detection

In this section, the parametric skin colour segmentation approach implemented in the context of this project is presented. This approach is based on the work of Terrillon *et al.* [10, 11] which uses a skin chrominance model built over the TSL (Tint-Saturation-Luminance) colour space. It assumes that the chrominance of Caucasian skin can be modeled by an unimodal elliptical Gaussian joint probability density function. Once the model is built, the Mahalanobis metric is used to discriminate between skin and non-skin pixels.

4.3.1 Chrominance model of Caucasian human skin

In order to build a chrominance model of human skin, the TSL colour space has been selected. This space provides robustness to illumination variation because it efficiently separates the chrominace component –tint and saturation– from the luminance one. Besides, it provides a confined and therefore easily to model skin colour distribution. Fig.5 shows the cumulative histogram obtained from the TS values of the skin colour pixels manually segmented of a set of 108 images. It must be noted that the luminance component of the TSL colour space has been removed and it can be also appreciated as the distribution of the skin colour is confined.

We assume that the distribution shown in Fig.5 can be modelled by an unimodal elliptical Gaussian joint probability density function given by

$$p[\bar{X}(i,j)/W_s] = (2\pi)^{-1}|\bar{\bar{C}}_s^{\frac{-1}{2}}| \exp\left[\frac{-\lambda_s^2(i,j)}{2}\right] \tag{1}$$

where $X(\bar{i},j) = [\bar{T}(i,j)\bar{S}(i,j)]^T$ represents the random measured values of T (tint) and S (saturation) of a pixel with coordinates (i,j) in an image. W_s is the class describing the skin colour. $\bar{\bar{C}}_s$ is the covariance matrix of the skin colour distribution:

$$C_s = \begin{bmatrix} \sigma_{T_s}^2 & \sigma_{TS_s} \\ \sigma_{TS_s} & \sigma_{S_s}^2 \end{bmatrix} \tag{2}$$

and $\lambda_s(i,j)$ is the Mahalanobis distance from vector $\bar{x}(i,j)$ to the mean vector $\bar{m}_s = [m_{T_s} m_{S_s}]^T$ obtained from the skin colour distribution. Equation (1) means that the probability of a pixel to be a skin colour pixel depends on the covariance matrix of the skin colour distribution as well as on the Mahalanobis distance between the pixel colour and the mean colour of the skin

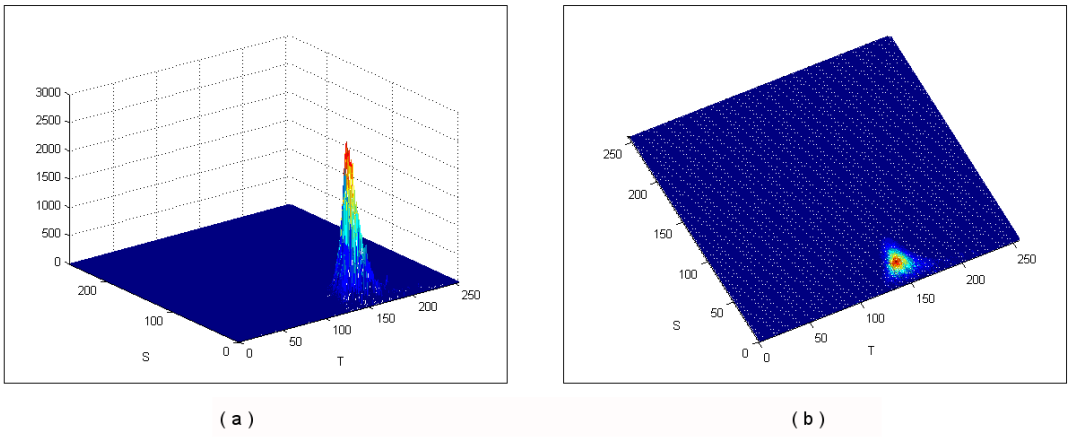


Figure 5: a) Histogram of the skin colour distribution; b) top view of the histogram.

distribution. Therefore, the larger $\lambda_s(i,j)$, the lower the probability that the pixel be a skin pixel.The Mahalanobis distance is given by

$$[\lambda_s(i,j)]^2 = [\bar{X}(i,j) - \bar{m}_s]^T \bar{\bar{C}}_s^{-1} [\bar{X}(i,j) - \bar{m}_s] \tag{3}$$

Equation (3) defines elliptical surfaces in chrominance space of scale $\lambda(i,j)$, centered about $\bar{m}_s$ and whose principal axes are determined by $\bar{\bar{C}}_s$.

Equations (1) and (3) show that the skin colour chrominance model is whole described by $\bar{m}_s$ and $\bar{\bar{C}}_s$. The values obtained for the skin colour distribution shown in Fig.5 were the following:

$$\bar{m}_s = [\ 149.0228 \quad 23.0944\ ] \tag{4}$$

$$\bar{\bar{C}}_s = \begin{bmatrix} 0.0058 & 0.0009 \\ 0.0009 & 0.0094 \end{bmatrix} \tag{5}$$

4.3.2 Skin colour segmentation

Once the parameters of the model have been computed, it can be used to extract skin colour regions from real images. The process to segment an input image is the following:

1. The RGB input image is transformed in a TSL image.

2. $\lambda(i,j)^2$ is computed for each pixel of the input image using equation (3). Each value is compared with a threshold λ_T^2.

3. A value of 1 is assigned to pixel (i,j) if $\lambda(i,j)^2 \leq \lambda_T^2$. Otherwise, pixel (i,j) is set to 0.

The output of the skin colour segmentation algorithm is a binary image where the skin colour pixels are set to 1 and non-skin colour pixels are set to 0.

The threshold λ_T^2 depends on the used camera and can be computed studying the percentage of false positives provided by the segmentation process. In our case, a range of false positives between 10% and 28% produces $\lambda_T^2 \in [6..10]$. Finally, in this work we propose to include two new thresholds in the model. These thresholds avoid that grey and black pixels will be included in the computed skin colour distribution. Thus, grey pixels are characterised by a small saturation value and a random tint value, and they can be removed from the skin colour distribution if only pixels with a saturation value higher than S_T are considered as skin colour pixels. On the other hand, black colour is only characterised by a low L value. Then, black regions can have random values in T and S. As the model only takes into account T and S values, then it is possible to classify a black pixel like a skin coloured pixel. To avoid that, only pixels with $L > L_T$ are included in the model.

One of the main problems of the skin colour detector is that these three thresholds must be adjusted to the finally employed camera. Thus, Fig. 6 shows some results obtained where parameters have been experimentally adjusted.

4.4 Face detection

A first step for any face processing system (i.e. face recognition or facial expressions identification systems) is to detect if one or more faces are presented in the image and to compute their locations. Given an arbitrary image, the goal of a face detection system is to determine whether or not there are faces in the image and, if present, return the image location and area of each face [12]. The existence of factors which can modify the appearance of a face in the image makes face detection a challenging task. Some of these factors are the presence of structural components in the face as beard, mustache, hat or glasses, the pose and orientation, the facial expression, the variations in the illuminance, oclussions and noise.

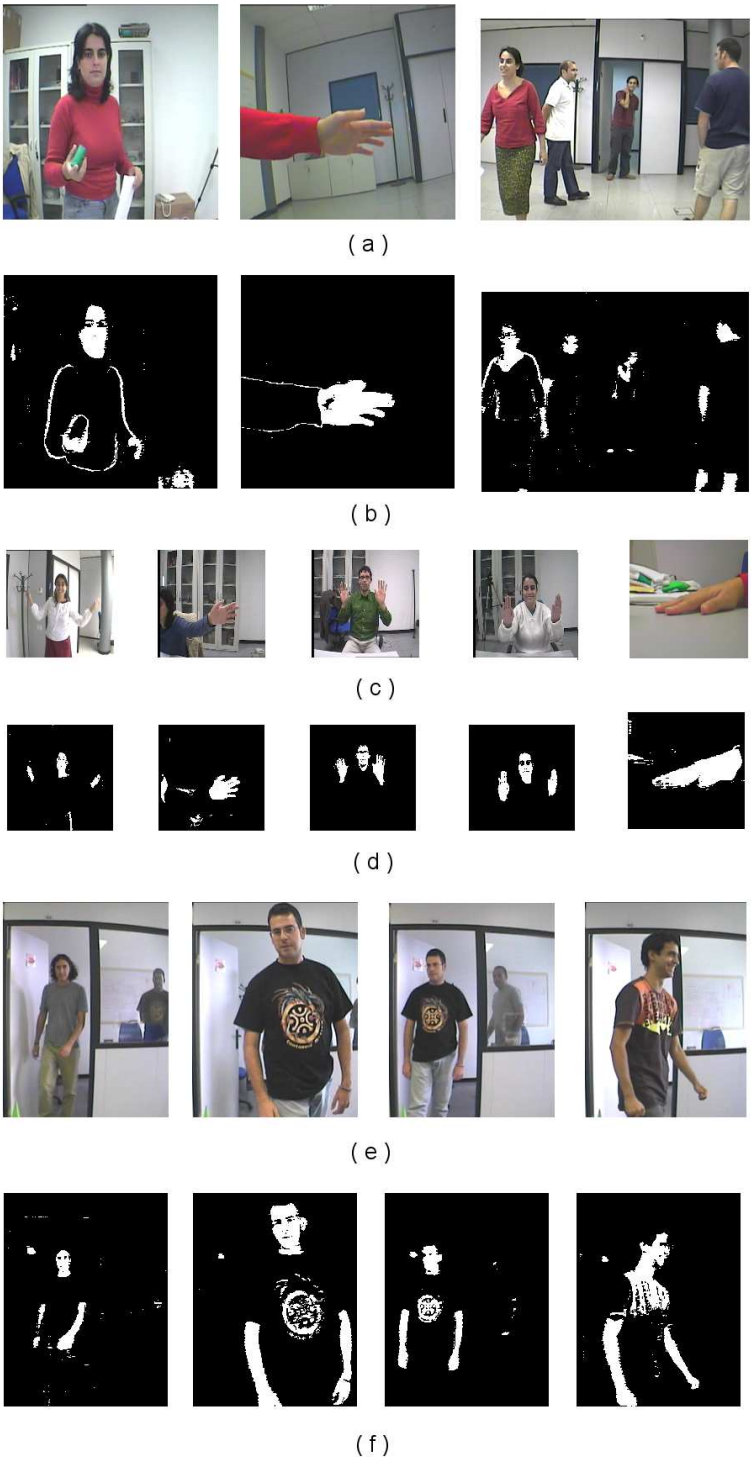


Figure 6: a-c-e) Original images; b-d-f) skin detection results $\lambda_T^2 = 10.0$ $S_T = 10$ $L_T = 80$.

In this project, a feature-based method for face detection is proposed (see [12] for alternative approaches). This method is based on the previous work of Viola and Jones [13], which uses Haar-like features to detect faces. Haar-like features encode the existence of oriented contrasts in the input image. This method has proven to be very fast (15 frames per second in a conventional desktop). Two of the main characteristics of the Viola and Jones's method, that are exploited in the work proposed here, are the following:

- The use of a set of features which are reminiscent of Haar Basis functions. In order to compute these features very quickly at many scales, they introduce a new image representation called *integral image*. It can be computed from an image using a few operations per pixel. Once computed, any of the Haar-like features can be calculated at any scale or location in constant time.

- A simple an efficient classifier is used to select a small number of important features from the huge amount of potential ones. This classifier is built using the AdaBoost learning algorithm [14].

Although the proposed algorithm is based on the key ideas of [13], a main contribution is presented. While Viola and Jones compute the Haar-like features over the whole image, we propose to previously detect skin colour regions in the input image and then to compute the Haar-like features only in the set of skin colour regions where a face is probably located. In order to select these potential *"face regions"*, a set of tests is computed over each skin region. Therefore, the face detection method proposed has two main steps:

1. Potential face regions of the input image are detected. This step can be subdivided into two stages: first, the skin colour pixels of the input image are detected and grouped into connected regions. Second, a set of structural tests is applied to the previously detected skin regions in order to discard the regions that clearly are not a face.

2. The remaining regions are classified as face or not face using the method proposed by Viola and Jones [13].

Different modules perform the skin colour detection, structural-based filtering and classification. If the first task can be included in the pre-attentive stage of the vision system, the other two tasks will be conducted at the semi-attentive stage.

4.4.1 Potential face regions extraction

The first step of the proposed face detection system is to compute the skin colour pixels of the input image. This resulting "skin image" is eroded and dilated in order to remove small noisy regions. Then, the connected skin colour regions are computed using a region labelling algorithm. Once the skin pixels of the skin image are grouped into connected regions, those whose dimensions are clearly not the dimensions of a face are discarded. In order to do that, four different tests are applied to the connected skin regions:

1. Test of minimum and maximum area: the skin regions whose area is less than the 1% of the total area of the input image are discarded. The skin regions whose area is higher than the 80% of the total area of the input image are discarded.

2. Test of elongated regions: each skin region whose bounding box height is less than the 40% of its bounding box width is discarded. Each skin region whose bounding box width is less than the 40% of its bounding box height is discarded.

3. Test of sparse regions: each region whose area is less than the 50% of the area of its bounding box is discarded.

4. Test of proportion: if the $height/width$ proportion of a region is higher than 1.6, the height of the region is reduced until $(height/width) < 1.6$.

All the previously used thresholds have been empirically obtained and they can be changed in order to control the flexibility of the tests. Fig. 7 shows the regions obtained after applying the tests to the labelled images.

Once the previously explained set of regions is discarded, the remaining regions are used to build 24x24 images which will be inputs of the classification algorithm. This algorithm is employed to discriminate between face and non-face regions.

4.4.2 Classification algorithm

The classifier is built using the Adaboost learning algorithm [14] which selects a small set of critical face features from a large set of features. The used features are the Haar-like features [15]. This classifier has been proposed by Viola and Jones [13].

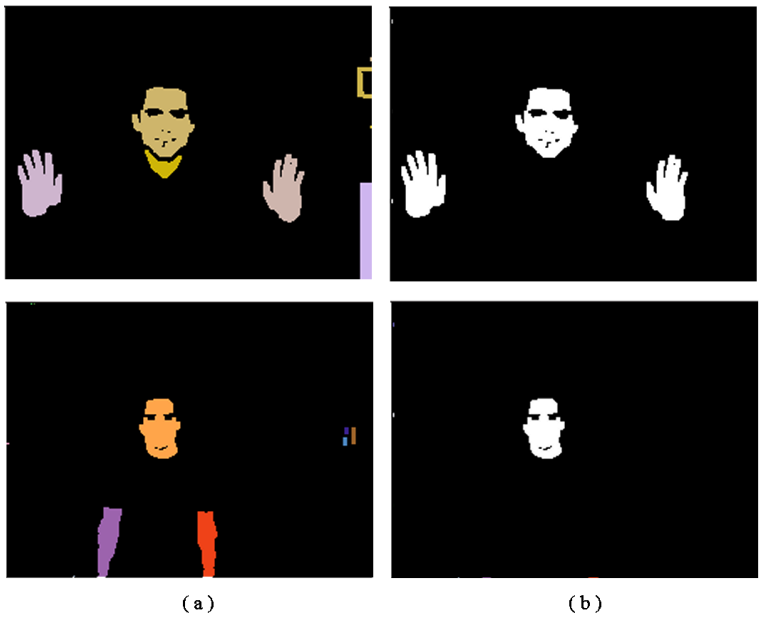


Figure 7: a) Labelled images; b) remaining regions after the tests.

Fig.8 shows three of the used features. In all examples, the sum of the pixels which are within the white rectangles are subtracted from the sum of the pixels in the grey rectangles, providing the feature value. These feature values can be computed very quickly from the *integral image* [13]. The integral image is an intermediate representation for the image which at location (x, y) contains the sum of the pixels above and to the left of (x, y) inclusive. Using the integral image it is possible to compute the sum of the pixels within any image rectangle with only four memory accesses [13]. Therefore, a two-rectangle feature is computed with six memory accesses. A three-rectangle feature needs 8 memory accesses and nine memory accesses are needed to compute a four-rectangle feature (see Fig. 8).

From each subimage, an excessively high number of features can be computed. To select a small set of critical features (T) and to train the classifier, the Adaboost algorithm can be used. The Adaboost learning algorithm consists of T weak classifiers (one for each feature) which are combined to form a strong classifier. Each weak classifier is designed to select the single rectangle feature which best separates the positive and negative examples. For each feature, the weak learning process determines the optimal threshold

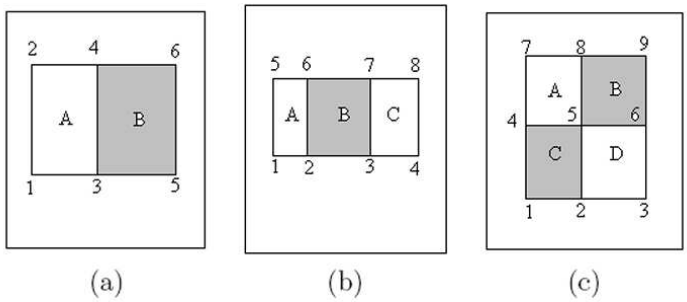


Figure 8: Memory accesses needed to compute: a) a two-rectangle feature; b) a three-rectangle feature; c) a four-rectangle feature. For example the two-rectangle feature is computed as: $B - A = (5-6-3+4)-(3-4-1+2)$

classification function, such that the minimum number of training images are misclassified. Therefore, a weak classifier ($h(x, f, p, \theta)$) consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The process to select the optimum set of weak classifiers from the whole set of possible weak classifiers is detailed presented in [13]. Once this set of weak classifiers has been obtained, the final strong classifier is a lineal combination of them. It is represented using equation (7).

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

being $C(x) = 1$ when the input image is classified as face and 0 in another case.

4.4.3 Experimental results

In order to perform the training of the classifier, we have used a set of $N = 200$ positive (face) and negative (non-face) images. Specifically, 100 positive and 100 negative images have been used. The number of computed potential features has been $K = 108,241$. Although the total number of features is 134,736 in a 24x24 subimage, we have discarded some of them because their contribution to the training process is not important. The number of

training iterations and final features has been $T = 150$. This number has been empirically obtained.

The features obtained in the training process are used to detect the faces presented in real images. Fig. 9 shows the results of the face detection process in several images. The face detection process has proven to be very fast: 0,033 seconds (30 fps) with 192x256 images and 0,043 seconds (24 fps) with 256x320 images using a 2,4 GHz Pentium IV PC.

4.5 Face recognition

This Section presents the development, implementation and experimental results for a system named 'Automatic Face Learning for the Welcome Desk'. It is embedded in a scenario where a person approaches the welcome-desk to get his face learned. The welcome-desk is mainly a PC equipped with a fire-wire camera. The system asks for permission to include the user to the 'godfather' database. In case the user agrees some images of his/her face will be collected and stored including the user's name. The system will first execute the learning and afterward the classification algorithm.

4.5.1 Theory

Principal Components Analysis (PCA) Kirby and Sirovich demonstrated based on the Karhunen-Loe'ave transform (aka principal component analysis) that images of faces can be linearly encoded using a modest number of basis images [16]. Given a collection of n by m pixel training images represented as a vector of size m by n, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images is minimized. These eigenvectors are later known as Eigenfaces since these are simply the eigenvectors of the covariance matrix computed from the vectorized face images in the training set.

Turk and Pentland applied principal component analysis to face recognition and detection [17]. Similar to [16], principal component analysis on a training set of face images is performed to generate the Eigenfaces which span a subspace (called the face space) of the image space. Images of faces are projected onto the subspace and clustered. Similarly, nonface training images are projected onto the same subspace and clustered. Since images of faces do not change radically when projected onto the face space, while the

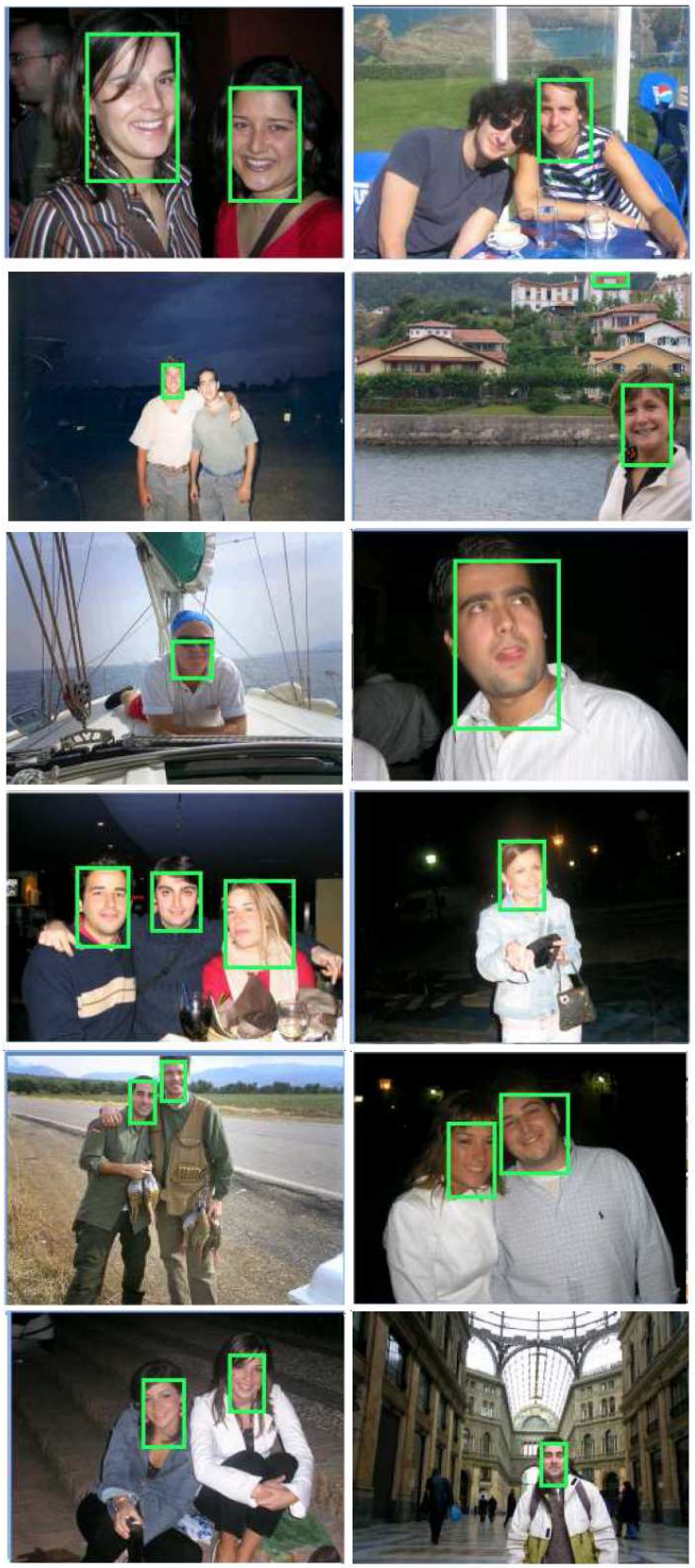


Figure 9: Face detection results

projection of nonface images appear quite different. To detect the presence of a face in a scene, the distance between an image region and the face space is computed for all locations in the image. The distance from face space is used as a measure of 'faceness' and the result of calculating the distance from face space is a 'face map'. A face can then be detected from the local minima of the face map. Many works on face detection, recognition, and feature extractions have adopted the idea of eigenvector decomposition and clustering.

Eigenfaces in Face Recognition The recognition task intents to extract the relevant information contained in a face image encoding it for future comparison with a also encoding models in a database. The simplest way to extract the information in a face is to capture the variation in a collection of face images. Given a training set of images we want to find the principal components of the distribution of faces, i.e. its eigenvectors. Together, these eigenvectors characterize the variation of a face image or a set of features for each face. Each eigenvector has a weight for an image and can be displayed producing a ghostly image called eigenface.

The face recognition system is based on eigenspace decompositions for face representation and modeling. The learning method estimates the complete probability distribution of the faces appearance using an eigenvector decomposition of the image space. The face density is decomposed into two components: the density in the principal subspace (containing the traditionally-defined principal components) and its orthogonal complement (which is usually discarded in standard PCA).

Suppose a face image consisting of N×N pixels, so it can be represented by a vector Γ of dimension N. Let $\{\Gamma_i | i = 1, ..., M\}$ be the training set of face images. The average face of these M images is given by

$$\Phi = \frac{1}{M} \sum \Gamma_i. \tag{8}$$

Then each face Γ_i differs from the average face Φ by Φ_i.

$$\Phi_i = \Gamma_i - \Phi, i = 1, ..., M. \tag{9}$$

A covariance matrix of the training images can be constructed as follows:

$$C = AA^T, \tag{10}$$

where $A = [\Phi_1, ..., \Phi_M]$. The basis vector of the face space, i.e., the eigenfaces, are then the orthogonal eigenvectors of the covariance matrix C. Finding the eigenvectors of the N×N matrix C is an intractable task for typical image sizes, hence, a simplified way of calculation has to be adopted [15]. Since the number of training images is usually less than the number of pixels in an image, there will be only $M - 1$, instead of N, meaningful eigenvectors. Therefore, the eigenfaces are computed by first finding the eigenvectors, $v_l(l = 1, ..., M)$ of the M×M matrix L:

$$L = A^T A, \tag{11}$$

The eigenvectors, $u_l(l = 1, ..., M)$, of the matrix C are then expressed by a linear combination of the difference face images, $\Phi_l(i = 1, ..., M)$, weighted by $v_l(l = 1, ..., M)$:

$$U = [u_1, ..., u_M] = [\Phi_1, ..., \Phi_M][v_1, ..., v_M] = A \times V. \tag{12}$$

4.5.2 Implementation

The system architecture consists of three main modules: face detection, learning and face recognition, and an initialization module, like shown in Fig. 10

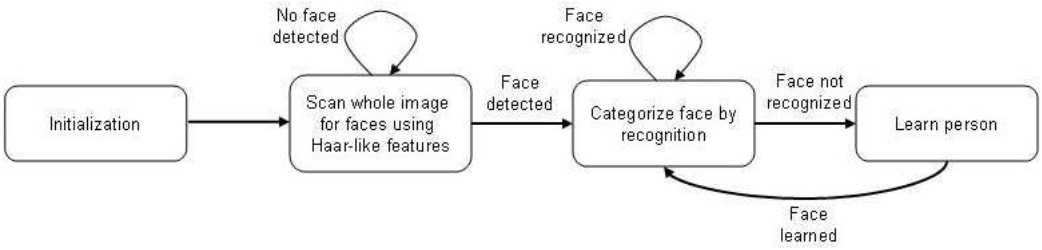


Figure 10: State diagram for detection, recognition and learning

In order to implement the Automatic face-learning for the Welcome Desk we use the OpenCV [18] library by Intel. This is a open source computer vision library providing a set of functions that might prove to be useful for our task.

Overview

Database During the recognition process a database of known people is used, with two directories for each person. Through the names of the directories we can infer the names of the people and their number. Each know person has a recorded sequence of 250 images of the upper half of the body (size 320x240 pixels, RGB). From this sequence 20 images of the face are created and stored (size 32x32 pixels, gray scale). The former is further on referred as the test set, the late as the training set.

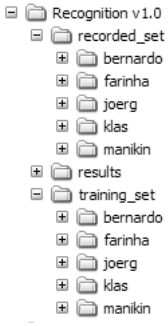


Figure 11: Directory structure of the database. The training_set folder contains a set of 20 face images per person. The recorded_set folder contains 250 images per person and along this report is referred as test set. In the results folder we store some important information collected during run time

Initialization As can be seen in Fig. 10 the system requires a initialization module, to collect some information to be used in the detection, recognition and learning modules. In Table 1 we describe the operations performed in the *Initialization* module.

Initialization

1. Read number and names of known people.
2. Read training sets.
3. Build eigen spaces using the function *cvCalcEigenObjects**.

4. Read thresholds θ_i.

* Function from openCV library [18].

Table 1: Operations performed in the *Initialization* module

Face detection The way of creating images of the face is the same through the whole program whether for learning or for recognition. From the input images (whether recorded or grabbed) the Haar-like feature detector will extract images of the face. Finally the images are scaled 32x32 pixels. These operations are performed by the Face Detection module, by using the function cvHaarDetectObjects from [18].

Recognition In the recognition process we use the previously scaled image to categorize a face. By projecting this image into the eigenspaces of the known people we get a projected image. Correlating these two images we get a distance measure, according to Eq. 13.

$$\varepsilon = \frac{\sum_{x,y} (Img2(x,y) - Img1(x,y))^2}{\sqrt{\sum_{x,y} Img2(x,y)^2 * \sum_{x,y} Img1(x,y)^2}} \tag{13}$$

Then, we make a conditional exclusion of the people through the thresholds, submitting the results to a filter which produces a result. The whole recognition procedure is described in the following table.

Recognition

1. (Input)
    1. A frame with face;
    2. Number of people in the database - NPEOPLE;
    3. Names of the people in the database - $names[i]$;

4. Threshold values for each person in the database - θ_i;
5. Results to filter - MAXFILTER;

2. (Projection and correlation)
    1. For i to NPEOPLE
        1. Create projected image of the detected face using the *cvEigenProjection** function;
        2. Correlate projected and detected image using the *cvMatchTemplate** function, resulting a distance measure ε, according to Eq. 13;
        3. Choose $\varepsilon_i < \theta_i$;
        4. $result[j] = \arg_i$ of $\min\{\varepsilon_i\}$;
        5. if $j{+}{+} =$ MAXFILTER goto Filtering;

3. (Filtering)
    1. For i to NPEOPLE
        1. For j to MAXFILTER
            1. if count $result[j] = i > \frac{MAXFILTER}{2}$
                - recognizedPerson = name[i];
            2. else
                - recognizedPerson = unknown;

4. (Output)
    - recognizedPerson;

* Function from openCV library [18].

Learning In the case that a person is unknown to the system we are able to add him/her to the database. Some interaction and time are required in this case. The user need to type his/her name and then the system will capture some images of the person, adding it to the database structure to building a test set and a training set. This is an important operation, once these sets of images are needed to learn the threshold for the new known person. The threshold is learned through the mean and standard deviation obtained recursively. Based on the mean

$$\mu_\varepsilon = \overline{\varepsilon} = \frac{1}{N}\sum_i \varepsilon, \tag{14}$$

we compute a recursive mean (Eq.15) using the ε (Eq. 13) resulting of the correlation between each image in the test set and its projection in the respective training set.

$$\mu_i = \frac{\mu_{i-1}}{i} + \frac{\varepsilon}{i} \tag{15}$$

The standard deviation is recursively obtained by using the last recursive mean (Eq. 15) and the actual ε value, according to Eq. 16.

$$\sigma_i = \frac{\sigma_{i-1}}{i} + \frac{(\mu_i - \varepsilon)^2}{i} \tag{16}$$

We finally calculate the threshold θ using the recursive mean (Eq. 15) and a factor k applied to the recursive standard deviation as described in Eq. 17.

$$\theta = \mu_\varepsilon + k \times \sigma_\varepsilon \tag{17}$$

In the following table we describe the whole procedure of the threshold learning.

Learning

1. (Input)
    1. N frames with a face from the person to learn - NFRAMES;
    2. Number of people in the database - NPEOPLE;

3. Names of the people in the database - $names[i]$;
4. Eigenspace of the person to learn;

2. (Learn threshold)
    1. For i to NFRAMES
        1. Load frame;
        2. Detect a face in the frame
        3. Extract, resize and convert the face to gray scale;
        4. Create projected image of the detected face using the *cvEigenProjection** function;
        5. Correlate projected and detected image using the *cvMatchTemplate** function, resulting a distance measure ε, according to Eq. 13;
        6. Create projected image of the detected face using the *cvEigenProjection** function;
        7. Correlate projected and detected image using the *cvMatchTemplate** function, resulting a distance measure ε, according to Eq. 13;
        8. Calculate recursively the mean μ_ε as described in Eq. 15.
        9. Calculate recursively the standard deviation σ_ε as described in Eq. 16;
    2. Calculate the threshold according to Eq. 17

3. (Output)
    - Write threshold to a file;

* Function from openCV library [18].

$$\sigma_\varepsilon = \sqrt{\frac{1}{N}\sum_i (\varepsilon_i - \mu_\varepsilon)^2} \tag{18}$$

4.5.3 Experiments

Overview Along this work we will test the behavior of our recognition system in different situations. To avoid some confusions let us, first, define the most common expressions used along the experiments description:

- **Training set** - set of images, pertaining to a defined subset, used to form the eigenspace.
- **Test-set** - set of images, pertaining to a defined subset, used as input.
- ***Named*-set** - set of images pertaining to the *named* subset.
- **Normal conditions** - set of predefined conditions used in some experiments, as described next:
  - The model is posing in frontal face.
  - The model background color is light gray.
  - The room illumination is composed of all ceiling lights on plus 2 halogen lights projecting also to the ceiling.
  - The camera was manually set using the setting from Table 2.

Camera settings					
brightness	320	auto-exposure	510	sharpness	100
blue/u	100	red/v	65	saturation	100
gamma	1	shutter	7	gain	150

Table 2: Camera setting along the experiments

Considering a space set of images, we have defined some particular conditions that may have heavy influence in the performance of our recognition system. Thus, we have defined a few subsets, represented for our model - the manikin, in the Fig. 12, used in the experiments. The subset manikin0 contains the normal conditions, as described before. In the manikin1 subset are

contemplated different face poses, i.e. this sub-set has faces rotated $\pm 30^0$ in a vertical axis. The manikin2 subset is related to variations in the background color. We have test 3 different background colors: black (manikin2b), gray (manikin0) and white (manikin2w). As a training set, for one person, have 20 face images, we defined a training set for manikin2 having 7 images with black background, 7 with white background and 6 with gray background. The manikin3 subset deals with different light direction and intensity.

The Fig.13 shows examples of distinct face images pertaining to each previously described subset. The represented images are extreme cases inside each subset. Paying attention to the different subsets (see Fig. 13) and to the Fig. 12 we can see that all them have at least one image taken in normal conditions, i.e, images b), e) and h) from Fig. 13 are similar to the images in the *manikin0* subset.

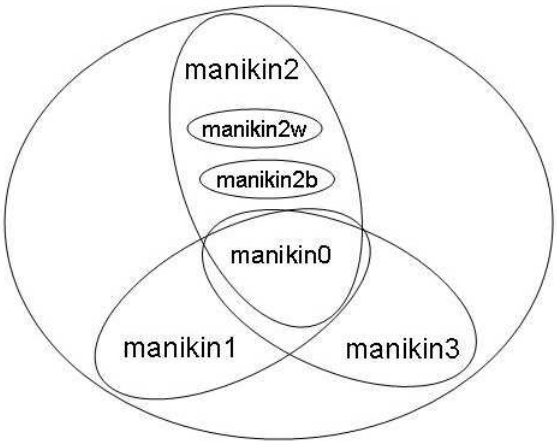


Figure 12: Face images set

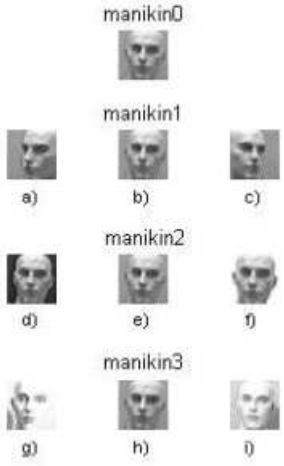


Figure 13: Set of images for the manikin

In the next list we present a brief description of ours experiments:

- **Experiments I** - Test the behavior of each test-set in a normal conditions and a specialized training set, evaluating its influence in the recognition system.
- **Experiments II** - Build an optimal training set based on the results

achieved in Experiments I. Test this training set using all available test-sets.

- **Experiments III** - Build training sets of different known people. Study the recognition system performance addressing the use of different thresholds definitions.

- **Experiments IV** - Study the thresholds calculation method concerning to the size of the training set and the test sequence.

Evaluation measurements To analyze the performance of the defined $\theta_{threshold}$ in the recognition process we have defined the tags **False Acceptance** (FA) and **False Rejection** (FR) as evaluation measurements. If a distance ε is below $\theta_{threshold}$ and is not a correct match we say that is a False Acceptance. On the other side, if a person is a correct match but the measure ε is above the define $\theta_{threshold}$ we say that is a False Rejection.

Experiments I Along this first experiment the face space set was constructed only with images from an individual - the manikin.

- **Test1: Testing manikin0-set on different training sets**. In the first test we have feeding in a manikin0 test-set in the different manikin training sets. Analyzing Fig. 14 we see that the mean of distances μ_ε is clearly small for the manikin0 training-set because we are comparing images from the same subset. For the training-sets manikin1 and manikin2 the mean μ_ε increases, suggesting that both the different facial poses and the background variations have great influence in the recognition process. The distance ε for manikin3 training set is similar to the achieved for the manikin0 training set.

- **Test2: Testing manikin1-set on manikin0 and manikin1 training sets**. In this section, the effect of different face poses is tested. If the training set is from the manikin1-set and using both manikin0 and manikin1 test-set, we achieve an low mean μ_ε, however the σ_ε dispersion is higher for the manikin0 training set (see Fig. 15). Along this test we saw that if the face pose is close to $\pm 30^0$ the distance ε grows a lot. This explains some dispersion σ_ε in the mean μ_ε.

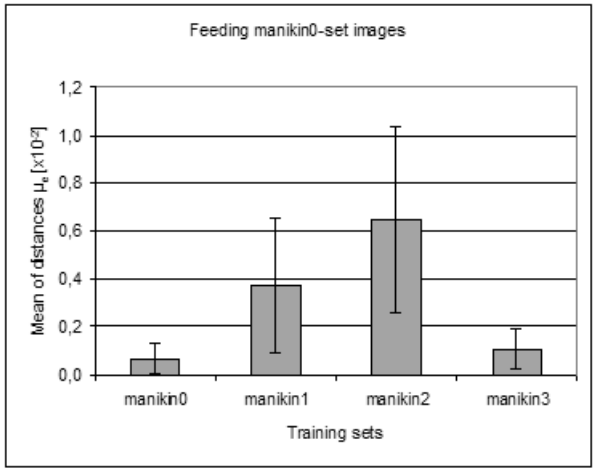


Figure 14: Mean of distances feeding in images from manikin0-set in different manikin training sets

- **Test3: Testing manikin2-set on manikin0 and manik training sets**. Face background plays an important role in our recognition process because entire information in the face image is used, without discarding any part of the image. High contrasts between the background of the training set and the test-set leads to a high value of the mean of distances μ_ε. As can be seen in Fig. 16 we achieve low mean μ_ε using a mix-background training set (manikin2), for all the different test-sets. However, if the test-set is from manikin2b or maninkin2w and the training set is from the manikin0-set the mean μ_ε increases quickly. Comparing the results achieved using the training set manikin0 and feeding in images from the manikin2b and manikin2w subsets we see a huge difference in the means μ_ε. To explain this we need to refer that the background color in the manikin0 subset is light-gray, hence, much more close to the white background color used in the manikin2w subset. The results from this test proof that the recognition process has some problems dealing with backgrounds color variations.

- **Test4: Testing manikin3-set on manikin0 and manikin3 training sets** The influence of the light direction and intensity are addressed in this test. We see, that, if the test-set was taken from then manikin3-set the mean of distances μ_ε is lower compared to a test-set taken from

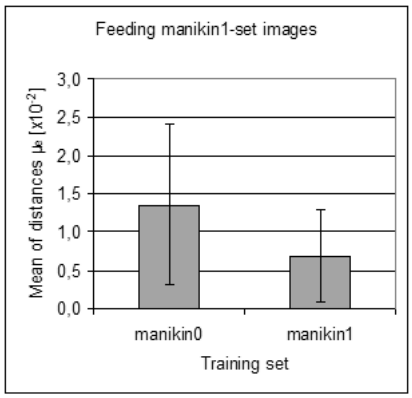


Figure 15: Mean of distances feeding in images with face pose variation (from manikin1-set) in different manikin training sets

the manikin0-set (see Fig. 17).

Experiments II Based on the evaluation of the results from Experiment I, we defined the composition of an optimal training set, shown in the Table 3, to use in this experiment. We are, now, interested to test the influence of variation in the training set images. In this experiment the training was

Nimages	**Type**	**Subset**
2	normal conditions	manikin0
4	left face	manikin1
4	right face	
3	black background	manikin2b
3	white background	manikin2w
4	different light	manikin3

Table 3: Composition of an optimal training set for the manikin

made with an optimal set from manikin, as described in Table 3. Then, we have feed in face images from our known subsets and evaluate the results.

The mean of distances μ_ε achieved with this training set is quite similar to the one using the specialized training set (in section Experiments I). However, the system still treats with some difficult different backgrounds. In the Fig. 18 we can compare the results for the mean of distances using different training sets.

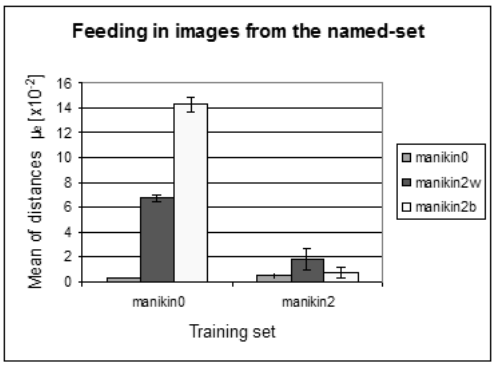


Figure 16: Mean of distances feeding in images with different background color (manikin2-set) in different manikin training sets

Experiments III In this experiment our first task is to build sets of face images from different known people. The set of images is captured using the conditions described next:

- The model face pose varies either in vertical or horizontal axis.
- The model background color is light gray.
- The room illumination is composed of all ceiling lights on.
- The camera was manually set using auto-exposure = 510, white balance = 100 and gain = 150.

Our interest is to define a threshold $\theta_{threshold}$ for the recognition system, using the mean of distances μ_ε, and evaluate the its performance. Training with the captured set of face images from one person, we, then, feed in test-sets from all the others known persons.

- **Test5: Define one threshold for everyone** To define the $\theta_{threshold}$ we use the furthermost μ_ε for a correct match (Bernardo matches Bernardo with a 5.74 mean of distances in Fig. 19) and the closest μ_ε for a wrong match (Joerg wrong matches Farinha with a 6.29 mean of distances in 19). The threshold $\theta_{threshold}$ is calculated like described in Eq. 19.

$$\theta_{threshold} = 6.29 - \frac{6.29 - 5.74}{2} = 6.015 \tag{19}$$

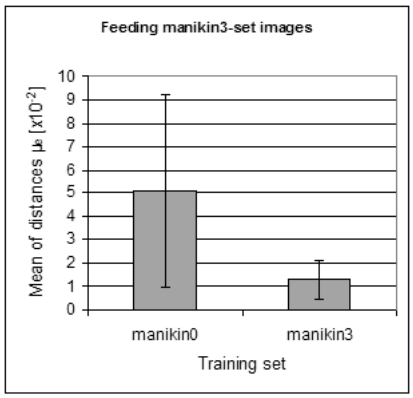


Figure 17: Mean of distances feeding in images with different light intensity and direction (from manikin3-set) in different manikin training sets

As can be seen in Fig. 19 there are situations where the defined threshold is extremely close to the achieved mean of distances for persons beside the correct one. In Fig. 19, if the training set is from Farinha or from Joerg set of images, the candidates to be a correct match are Farinha, Joerg and Manikin. In the table 4 we present the False Acceptance rate using the threshold calculated in Eq. 19. In the main diagonal of the same image we can see the recognition rate an infer the False Rejection rate by substraction that value to 100%. Once that Bernardo's μ_ε was used to calculate the threshold his recognition rate is quite low.

		Test set			
		Joerg	Bernardo	Manikin	Farinha
Training set	Joerg	98.4%	0%	21.2%	34%
	Bernardo	0%	52.4%	0%	0%
	Manikin	0%	0%	100%	0%
	Farinha	53.6%	0%	0%	97.6%

Table 4: False acceptance and recognition rate for known people using one threshold.

As can be seen in table 4 in some cases we have a strong False Acceptance, i.e. 53.6% times Joerg is accepted as Farinha and sometimes the Manikin and Farinha are also accepted as being Joerg. In Fig. 20

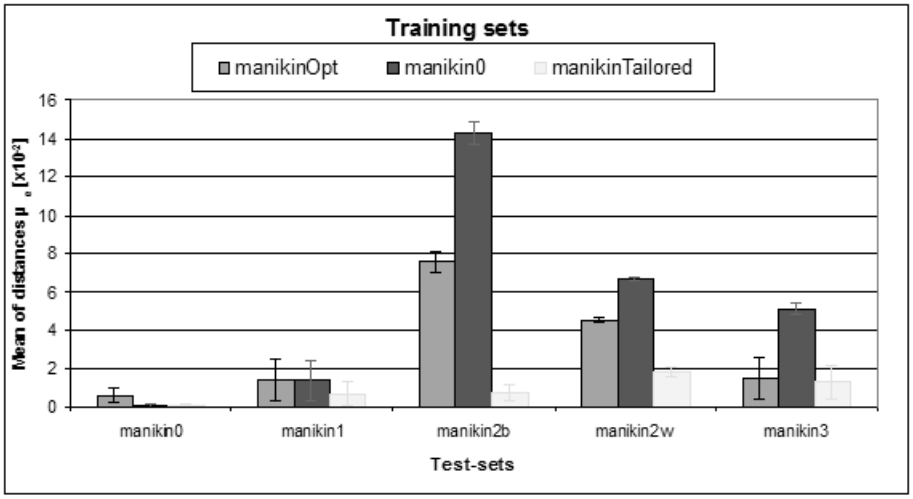


Figure 18: Comparison of the mean of distances among an optimal, normal conditions and tailored training sets using the manikin as model

we present the distances ε for Joerg and Farinha when we are trying to recognize Farinha. From that graph we clearly see that we need to decide among people with distances ε below the threshold. Another drawback of the use of one threshold for all people is the fact that sometimes False Acceptances occur simultaneously with False Rejections and also that the distance for False Acceptances are lower than distances for rights matches, e.g. in Fig. 20 there are 5.6% times that Joerg distance is better than Farinha distance.

- **Test6: Define one threshold for each known person** Based on the mean of distances μ_ε and in its variation σ_ε, showed in Fig. 19, we define a $\theta_{threshold}$ for each person, truncating the result of the addiction of μ_ε, Eq.14 and the positive variation of σ_ε, Eq. 18.

$$\theta_{threshold} = \mu_\varepsilon + \sigma_\varepsilon \tag{20}$$

In the Table 5 we present the $\theta_{threshold}$ to use in this test.

After after calculate the $\theta_{threshold}$ we follow the same procedure used in Test5, i.e, the training set is made with the captured set of face images from one person, and then, we feed in test sets from all the others known persons.

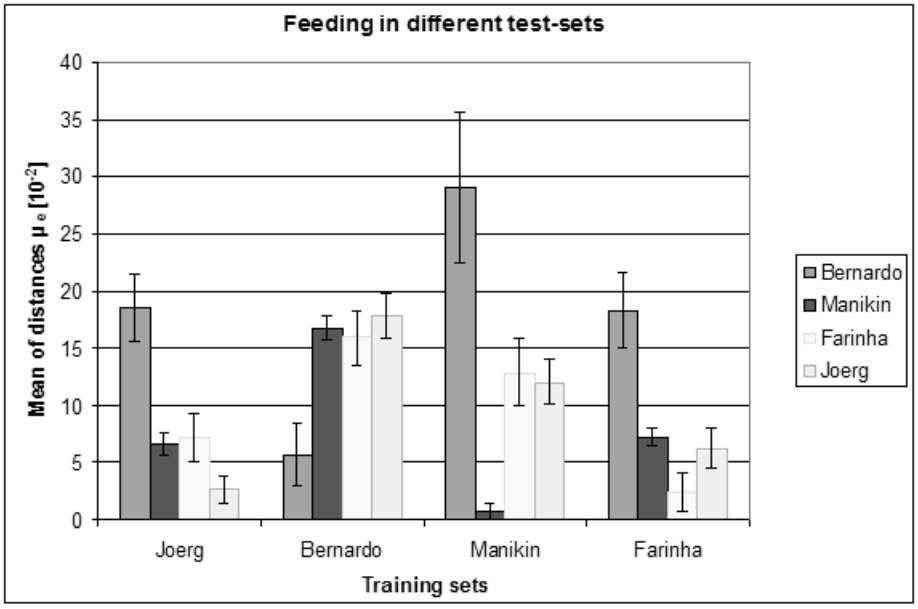


Figure 19:

Person	Bernardo	Farinha	Joerg	Manikin
$\theta_{threshold}$	0.08	0.04	0.04	0.02

Table 5: Threshold for each person in Test6

The reached results improved a lot, considering the False Acceptance rate from Test5, but the recognition rate has decreased (see table 6) for the majority people. This is explained by the rules applied to determine the thresholds. Paying attention in the $\theta_{threshold}$ calculation method we see that forcing this value to be great, the recognition rate can be improved, however, the False Acceptance rate will also increase.

In Fig. 6 we have presented the recognition rate using a fixed threshold value for each person. From now on, we will use a $\theta_{threshold}$ for each person, but, instead of use a fixed value we will use a value calculated using the Eq. 20.

Experiments IV After evaluating the performance of the threshold, calculated in different manners, it becomes interesting to see what happens varying the number of face images in the training set. Following this idea,

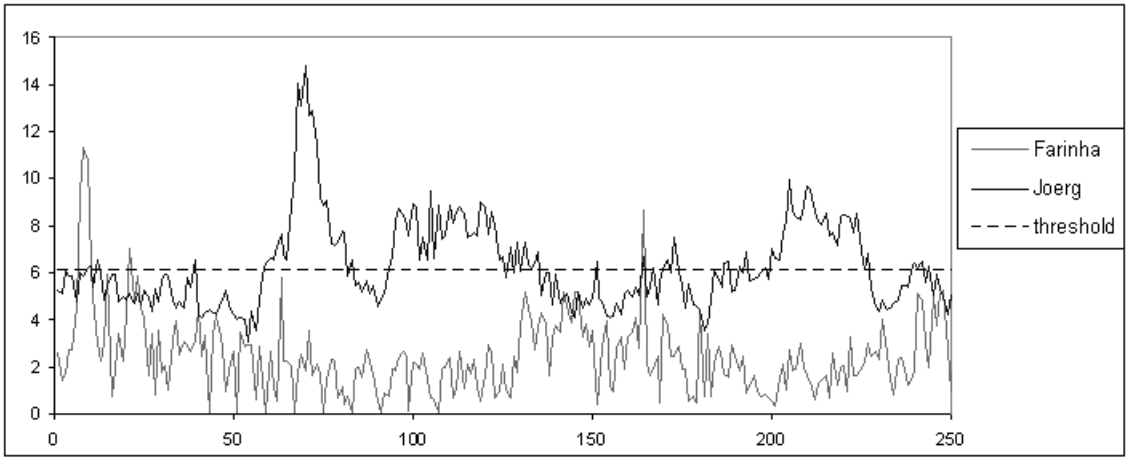


Figure 20: False acceptance and recognition rate for known people using different thresholds for each person.

		Test set			
		Joerg	Bernardo	Manikin	Farinha
Training set	Joerg	88.4%	0%	0%	0.8%
	Bernardo	0%	77.2%	0%	0%
	Manikin	0%	0%	98%	0%
	Farinha	1.6%	0%	0%	87.2%

Table 6: False acceptance and recognition rate for known people using one fixed threshold for each person.

we choose two known persons in our database, those that get the highest and the lowest thresholds, i.e. Bernardo and Manikin, respectively, and we evaluate the influence of the number of images in their threshold. It's also interesting, for us, to get a comparison measure for the recognition rate, based on the False Rejections occurrence.

The procedure of this experiment is simple.

Each time, we choose a set of 10, 20 or 50 images for each person, and then we learn the threshold, as described in section 3, using all the other images as test-set. The chosen images for the training set contain as much face pose variations as it is possible. In Fig. 21 we show an example of a training set.

In Fig. 22 we present the learned thresholds for the people in test, using

Figure 21: Set of 20 face images from Bernardo used as training set

different number of images in the training set. The thresholds we get for the Manikin varies slowly. Bernardo's threshold decreases quickly when we increase the number of images in the training set.

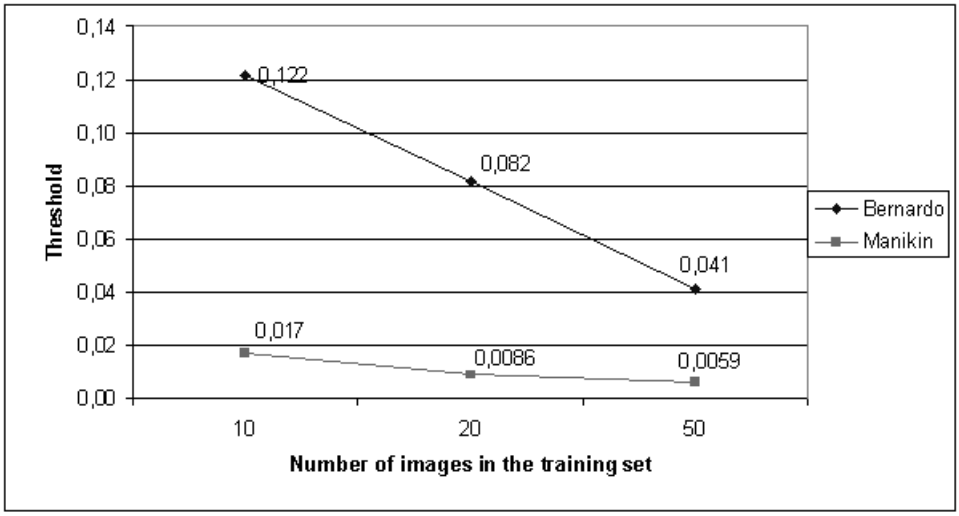


Figure 22:

4.6 Attention mechanism

As mentioned in Section 4.2, the visual perception system of a social robot should imitate the ability of natural vision systems to select the most salient information from the broad visual input. The use of attention to reduce the amount of input data has two main advantages: i) the computational load of the whole system is reduced, and ii) distracting information is suppressed. An attention mechanism is central to a system requiring a selection of the relevant information on which the system activities are based.

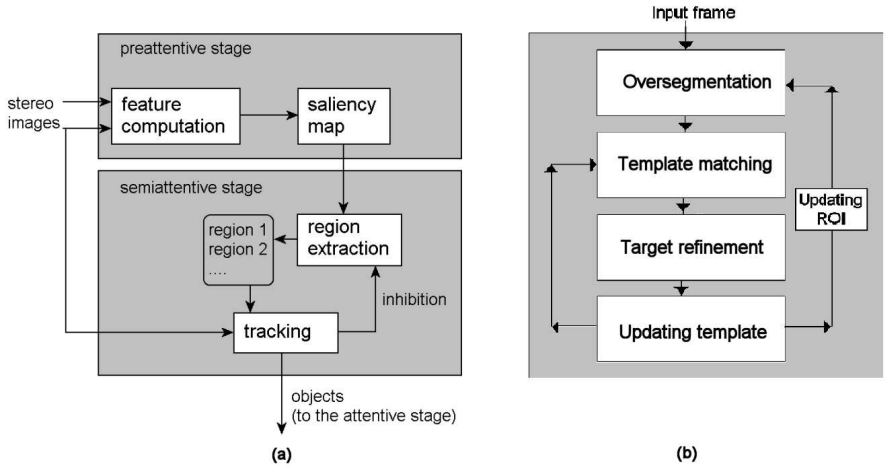


Figure 23: a) Overview of the proposed attention mechanism and b) overview of the tracking algorithm

In this project, a general purpose attention mechanism based on the feature integration theory has been implemented. It is capable of handling dynamic environments, and detecting objects of interest, e.g. human faces or hands, in a fast way. This mechanism integrates bottom-up (data-driven) and top-down (model-driven) processing. The bottom-up component determines and selects salient image regions by computing a number of different features. The top-down component makes use of object templates to filter out data and only track significant objects. Fig. 23.a shows the overview of the proposed architecture. This section is centered in the task-independent stage of a feature integration approach.

4.6.1 Pre-attentive stage

The proposed attentional mechanism uses a number of features computed from the available input image in order to determine how interesting a region is in relation to others. These features are independent of the task and they allow to extract the most interesting regions of the image. Besides, they allow to distinguish locations where a human may be placed. The chosen features are colour and intensity contrast, disparity and skin colour (see Fig. 24). In the final version of the algorithm, the Small Vision System (SVS) provided by Videre Design (www.videredesign.com) has been employed to extract an

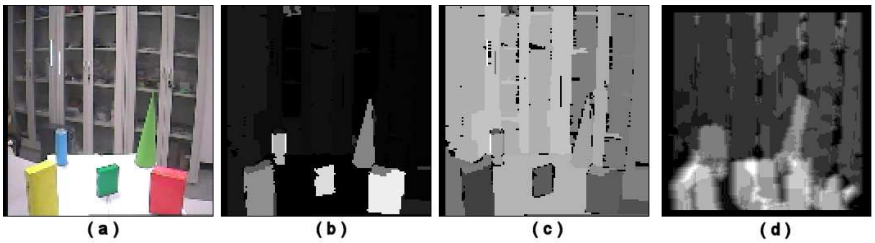


Figure 24: Colour and intensity contrast computation: a) Left input image; b) colour contrast saliency map; c) intensity contrast saliency map and d) disparity map

accurate disparity map. SVS is a set of library functions which implement stereo algorithms. The disparity map is computed using a correlation-based algorithm.

Attractivity maps are computed from these features, containing high values for interesting regions and lower values for other regions in a range of [0...255]. The integration of these feature maps into a single saliency map allows to determine what regions of the input image are the most interesting. Other features can be easily added without changes in the following steps.

Similarly to other models [19][5], the saliency map is computed by combining the feature maps into a single representation. A simple normalized summation has been used as feature combination strategy because, although this is the worst strategy when there are a big number of feature maps [20], it has been demonstrated that its performance is good in systems with a small number of feature maps. Fig. 25.b shows the saliency map associated to Fig. 25.a.

4.6.2 Semi-attentive stage

Once the saliency map is calculated, it is segmented in order to obtain regions with homogeneous saliency. Among the set of obtained regions, only big enough regions with a high saliency value are taken into account. Thresholds to select if a region can be considered as a region of interest has been empirically obtained.

A general problem in attention mechanisms is to avoid revisiting or ignoring salient objects of the image when the system is working in a dynamic environment with moving objects. To solve this problem, it is necessary to

include in the system a mechanism to avoid extracting the same objects in different frames, although they will be in different positions in the images. The way to solve the problem of revisiting or ignoring objects is called "inhibition of return" and the proposed attention mechanism implements it by including an algorithm which tracks the objects extracted from the scene. This tracking prevents the attention mechanism from wrongly identify them as new objects.

The tracking algorithm is based on the Bounded Irregular Pyramid (BIP) [2]. It permits to track non-rigid objects without a previous learning of different object views in real time. To do that, the method uses weighted templates which follow up the viewpoint and appearance changes of the objects to track. The templates and the targets are represented using BIPs.

The most salient regions obtained by segmentation of the saliency map are directly related to homogeneous colour regions of the segmented left input image. These homogeneous colour regions are the targets to track. Fig. 25.c shows the selected targets associated to the saliency map in Fig. 25.b. Once the targets are chosen, the algorithm extracts its hierarchical representations. Each hierarchical structure is the first template $M_r^{(0)}$ and its spatial position is the first region of interest $ROI_r^{(0)}$, where $r \in [1...N]$ and N is the number of salient regions to track. The main steps of the proposed tracking algorithm (Fig. 23.b) are detailed explained in [21].

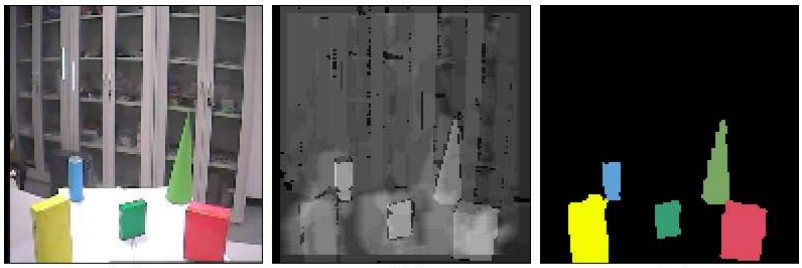


Figure 25: Saliency map computation and targets selection: a) Left input image; b) saliency map; and c) selected targets

4.6.3 Experimental results

The above described attentional scheme has been examined through experiments which include humans and objects in the scene. Fig. 26a shows a sample image sequence seen by a stationary binocular camera head. Every 10th frame is shown. All salient regions are marked by black and white bounding boxes in the input frames. It must be noted that the activity follows the objects closely, mainly because the tracker works with the segmented input image instead of working with the saliency image. Furthermore, the tracking algorithm prevents the related object templates from being corrupted by occlusions. Fig. 26b presents the saliency maps after inhibiting the regions which have been tracked in each frame. This inhibition avoids that the region extraction process extracts regions that have been already extracted in previous frames. It can also be observed how the mechanism follows appearance and view point changes of the salient objects.

4.7 Vision-based human motion capture

In this section, the implementation of a real-time human motion capture system based on computer vision is presented. The goal of this module is to extract the upper-body movements of a person without using any beacons or markers, using only two stereo cameras. The key idea behind this system is the assumption that in order to track the global upper human body motion, it is not necessary to capture with precision the motion of all its joints. Particularly, in this work only the movement of the head and hands of the human are tracked, because they are the most significant items involved in the human-to-human interaction processes. These are modeled by weighted templates that are updated and tracked at each frame using the previously mentioned hierarchical tracking approach. Besides, the silhouette information is processed to provide approximated positions for the elbows and an overall body orientation. The pose of the joints is then extracted through the use of a kinematic model of the human to track. It is also assumed that the human motion speed is bounded and that the pose of the different items to track is related to its last detected pose. By assuming this important constrains, the proposed system can estimate upper-body human motion at 25 frames per second in a standard PC.

An overview of the proposed system is shown in Fig. 27. The system has two main modules: a vision module and a joint angle extraction module.

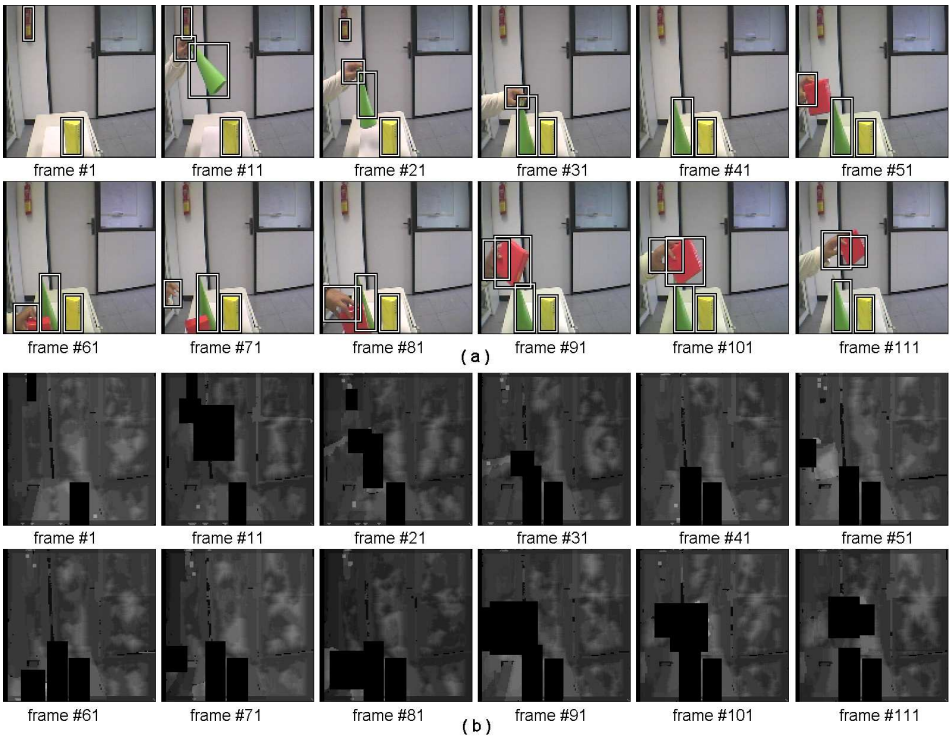


Figure 26: Example of selected targets: a) left input images; and b) saliency map associated to a)

The vision module extracts the 3D coordinates of the head and hands of the human using the attention mechanism previously explained which includes the hierarchical tracking algorithm in its semi-attentive stage. These 3d coordinates are used by the model-based joint angle extraction module located at the attentive stage to compute the pose of the upper-body joints by means of a kinematic model and a inverse kinematics algorithm.

4.7.1 Vision module

The main stage of the vision module is the attention mechanism previously explained in section 4.6. For this application only two features of the attentional mechanism are relevant: skin colour and disparity. That is, only skin coloured regions are used by this attentive module.

The disparity map is processed in order to extract the silhouette of the

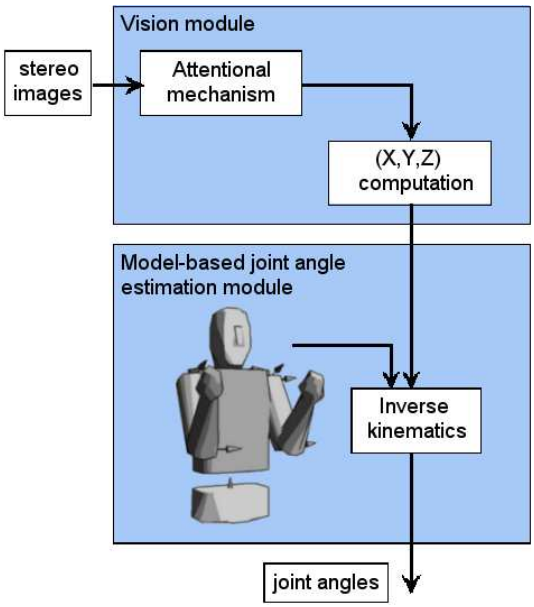


Figure 27: Overview of the proposed human motion capture system

person. To do that, the face detection algorithm presented in section 4.4 is used to determine the position of the human face in the input image. The mean value of the disparity of the localized face is used as threshold to reduce the number of disparity values in the disparity map. That is, only a certain number of disparities over this reference and below it are taken into account, the rest of values are removed from the map. This filtering is based in the fact that the maximum distance between the head and one hand of the same person is determined by the length of a stretched arm. We consider this length not to be superior to one meter. Thus, all disparities over this threshold are discarded. The result of this first filtering process is shown in Fig. 28c. Once this new map is obtained, the silhouette of the person is extracted using connected components (Fig. 28d). The hands of the person are determined as the biggest skin colour regions located inside of the silhouette. These hands and the face are the extracted salient regions which are tracked by the hierarchical tracking algorithm included in the semi-attentive stage. Therefore, the attentional mechanism is able to compute in each frame the 2D position of the head and the hands and their disparity values, as shown in Fig. 28d, providing to the attentive stage the 3D position of these items.

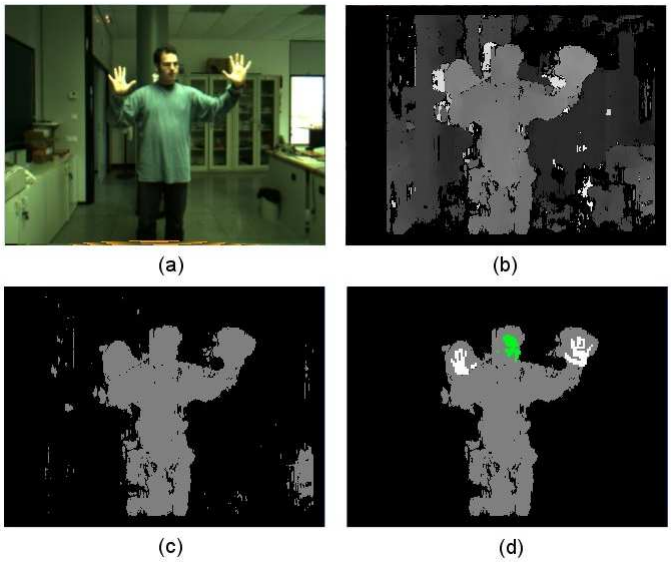


Figure 28: a) Left image of an input stereo pair; b) Disparity map; c) Relevant disparities (grey); and d) Extracted silhouette (grey), tracked face (green) and tracked hands (white).

4.7.2 Model-based pose generator

Our approach is exclusively based on the information obtained from the stereo vision system of the robot imitator. Thus, it is related to other experiments, e.g. the mimicking experiments shown by Sauser and Billard [22], but in our case external color marks are not employed. As explained above, the information extracted for each frame is restricted to 3D positions of head and hands. Wren and Pentland already developed a system to recover human motion from these limited cues, using physical constrains and probabilistic influences [23]. They also use a model to help in the tracking process by projecting 3D virtual blobs into 2D images taken with the stereo pair and improve pose estimation in a recursive scheme. The resulting system allows to track human upper-body movements at 30 fps, but it has to be manually initialized and requires several computers working on parallel due to its complexity.

Our system also uses a kinematic human model to translate 3D head and hands positions to a correct pose. But we base the translation in a fast analytic inverse kinematics algorithm running over a model that avoids

incorrect poses. This model filters tracked movements and provides, in real-time, a set of joint angles that conforms a valid human pose and preserves perceived 3D positions.

Model We have restricted ourselves to capture upper body motion. Thus, the geometric model contains parts that represent hips, head, torso, arms and forearms of the human to be tracked. Each of these parts is represented by a fixed mesh of few triangles, as depicted in Fig. 29. This representation has the advantage of allowing fast computation of collisions between parts of the model, which will help in preventing the model from adopting erroneous poses due to tracking errors.

Each mesh is rigidly attached to a coordinate frame, and the set of coordinate frames is organized hierarchically in a tree. The root of the tree is the coordinate frame attached to the hips, and represents the global translation and orientation of the model. Each subsequent vertex in the tree represents the three-dimensional rigid transformation between the vertex and its parent. This representation is normally called a skeleton or kinematic chain [24] (Fig. 29). Each vertex, together with its corresponding body part attached is called a bone. Each bone is allowed to rotate –but not translate– with respect to its parent around one or more axes. Thus, at a particular time instant t, the pose of the skeleton can be described by $\Phi^{(t)} = (R^{(t)}, \vec{s}^{\,(t)}, \phi^{(t)})$, where $R^{(t)}$ and $\vec{s}^{\,(t)}$ are the global orientation and translation of the root vertex, and $\phi^{(t)}$ is the set of relative rotations between successive children. For upper-body motion tracking, it is assumed that only ϕ needs to be updated –this can be seen intuitively as assuming that the tracked human is seated on a chair.

Fig. 29 shows the 3D kinematic model used in this system. It has four degrees of freedom (DOF) in each arm. Three of them are located in the shoulder, and one in the elbow. Model proportions and dimensions have been set to average human values, although they can be rescaling by the algorithm, as it will be discussed in next subsections.

Inverse kinematics As shown in Fig. 30, each arm is modelled with a two-bone kinematic chain. The parent bone corresponds to the upper arm and is allowed to rotate around three perpendicular axes. This provides a simplified model of the shoulder joint. $T({}^{w}_{1}R)$ is the local transformation between the upper-arm reference frame O_1 and a coordinate frame attached to the torso

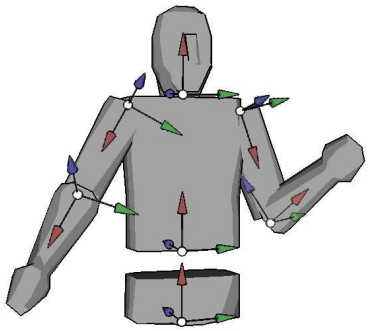


Figure 29: Illustration of the human upper-body kinematic model

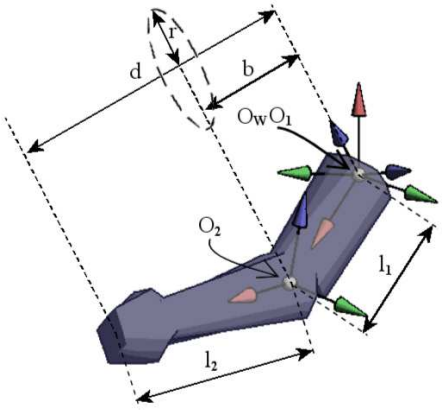
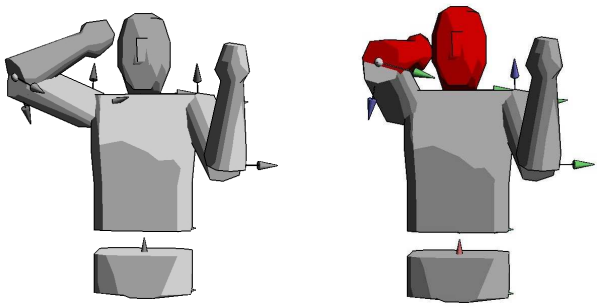


Figure 30: Kinematic model of the arm showing local coordinate frames and elbow circle (see text).

and centered at the shoulder joint w. The bone representing the lower arm is allowed to rotate around a single axis, corresponding to the elbow joint. $T({}^2_1R, {}^1\vec{l}_1)$ denotes the local transformation between the upper-arm reference frame O_1 and the lower-arm reference frame O_2, where ${}^1\vec{l}_1 = (0, 0, l_1)^T$, being l_1 the length of the upper-arm, and ${}^2_1R$ corresponds to the rotation θ_e about the elbow axis.

Given a desired position for the end-point of the arm at time instant $t+1$, ${}^w\vec{p}_d^{\,(t+1)}$, and given the rotation matrices ${}^w_1R^{(t)}$ and ${}^1_2R^{(t)}$ at the previous time instant t, the problem is then to find the updated matrices ${}^w_1R^{(t+1)}$ and ${}^1_2R^{(t+1)}$. A simple geometric method is employed to solve such problem. See



(a) (b)

Figure 31: RAPID collision detection: (a) Valid pose. (b) Collision.

[25] for further details.

Enforcement of joint limits and collision avoidance The proposed inverse kinematics method can obtain an arm pose that will put the hand of the model in the required position. The resulting pose must be analyzed in order to determine if it corresponds with a valid and natural body configuration. In this work we consider two limitations: a valid pose must respect joint limits and cannot produce a collision between different links.

- *Detection of joint limit violations.* Given the updated shoulder and elbow rotation matrices, it is necessary to extract joint angles from these matrices that correspond to the DOFs of the human model.

  This process is made by applying a parameterization change to rotation matrices. There is a direct correspondence between Denavith-Hartenberg (DH) [26] parameters and model joint angles, so the local axes referred angles are converted to DH parameters using an appropriate parameterization.

  Once the model DOFs are computed, the system can directly check if any of them lies beyond its limits.

- *Collision detection.* We use RAPID [27] as the base of the collision detection module. This library provides functions that can quickly and efficiently check collisions between meshes composed by triangles, such as the ones attached to the links in our model (Fig. 31).

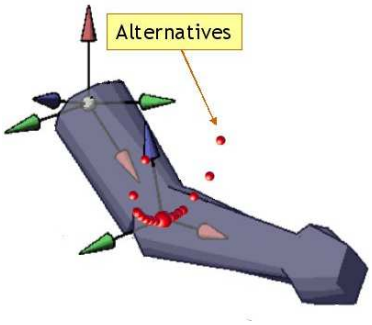


Figure 32: Alternative poses (red spheres) for a given elbow position.

Once the system detects an incorrect position (i.e. joint limit or collision), it follows these steps:

1. The system looks for alternative poses (i.e. different arm configurations). Imitation requires to place hands in certain coordinates, but the elbow is free to move in the circle presented in Fig. 30. Thus, alternative poses will preserve hand positions, but will move the elbow in this circle.

2. The motion of the arm should be as smooth as possible. Thus, alternatives should be more densely searched near the current elbow location. This is implemented by exponentially distributing the alternatives around the initial incorrect elbow position, as shown below:

$$\theta_{2i} = \pi \frac{1}{100^{\frac{(n-i)}{n}}}$$
$$\theta_{2i+1} = -\theta_{2i}$$
$$i = 0, 1, 2, ...(n-1) \quad (21)$$

where θ_{2i} and θ_{2i+1} correspond to two symmetric alternatives on the elbow circle with respect to the current pose, and $n = \frac{N}{2}$, being N the number of alternative poses checked when current pose is erroneous.

Fig. 32 shows alternatives given a certain pose. As required, alternative poses are placed on the elbow circle (Fig. 30) and are more deeply distributed near the current elbow position.

3. The system chooses the nearest valid alternative.

4. If there is no valid alternative, the arm remains in the last valid position.

The speed of the process depends on the number of alternatives it needs to check. A system using a correct number of alternatives should produce smooth movements and work in real-time even in the case in which all of them need to be checked.

The alternative evaluation module has been also used when the system is in a valid pose: in these cases, the two nearest alternatives to current pose are checked. If one of them locates the elbow in a lower vertical position, and does not produce violation of limits nor collisions, then the elbow is moved to that position. This allows the model to adopt more natural poses when possible.

Scaling the model to fit the human In order to coherently follow the movements of the human, the 3D model will be scaled to match demonstrator's height. The scale ratio will be the following:

$$ratio = \frac{height_{human}}{height_{model}} \tag{22}$$

In our implementation, the model height is 170 cm. The human height is determined by the 3D position of the human head, provided by the vision module.

Imitated motions are then easily normalized by simply re-scaling the model to its original size while preserving the joint angles sequence. In this way, motions of very different people can be analyzed and compared.

4.7.3 Experimental results

The proposed system has been tested using a STH-DCSG-VARX stereo system and the Small Vision System software, provided by Videre Design (www.videredesign.com). This architecture captures and preprocesses stereo pairs. The size of left and right images is 320x240. The disparity map has also a size of 320x240.

The face of the demonstrator is detected using the cascade detector described in section 4.4. The 3D virtual model used to reproduce perceived gestures is rendered and animated using OpenSceneGraph, an open source graphic engine available at www.openscenegraph.org.

The whole system runs on a 2 GHz Pentium IV computer using Linux operating system.

The experiments performed to test the Human Motion Capture System involved different demonstrators moving their hands in a non-controlled environment. The only imposed requisite was to wear long sleeves. Besides, as the stereo system has a limited range, the demonstrator was told to stay at more than 1.50 meters from the cameras. Fig. 33 show the results obtained by the proposed system at an average rate of more than 25 fps.

As shown in Fig. 33, the generated pose closely resembles the pose of the human demonstrator. The figure also shows that the disparity map computed by the Videre system presents some noise. This noise will introduce errors to the perceived depth of the head and the hands. The first versions of the proposed system tried to reduce those errors by averaging the disparity values of the skin pixels in the regions of interest. This lead to better results, but still the errors in some pixels, specially those located in the borders, tended to distort the results.

The current version of the system takes into account the confidence value for the disparity of each pixel provided by the SVS software to reduce the disparity noise. After several tests, it was decided that the best option was to simply take as disparity value for each region the one associated with the highest confidence into that region. Then, the disparity value and the pixel coordinates of the region centroid are used in combination with camera parameters to extract (X,Y,Z) coordinates of the head and the hands. In our system, the distance between the human and the cameras is around 170 cm. For these values, the theoretical depth resolutions for the SVS are under 1 cm - more precisely, 7 mm for a distance of 165 cm. But, as commented above, there are different sources that introduce errors in the disparity map and in the tracking algorithm. These errors reduce the effective resolution of the stereo system. In any case, the average error is less than 5 centimeters. This is a good enough result as the 3D model will help correcting incorrect poses.

Fig. 34 shows another sequence in which the demonstrator is performing fast and large movements. The hands, also, move near the body in some occasions. Still, the model is able to find a valid and natural pose in these situations, although sometimes the position of the elbow differs respect to human pose, as in the second frame.

When there is a substantial reduction in the size of the tracked region, there are less chances of having high confidence disparity values in the region.

This was the case in the third frame of Fig. 34, resulting in a pose which is different to the demonstrated one. Our future work will have to focus on this issue and improve results in this situation, using information about previous frames to filter the current position.

4.8 Navigation

4.8.1 Obstacle Avoidance Methods

Navigation in dynamic and real-world environments is a difficult and challenging task that is considered as a essential problem to resolve for human-robot interaction. Basically all these environments are characterized by their complex structure and the aleatory movement of humans and objects in them and around the robot. Thus the mobile agent has to avoid collisions with these different obstacles while still reaching its target position in a fast and efficient way.

Currently, obstacle avoidance is treated by different methods that can be grouped in two broad categories: *global* algorithms and *local* (or reactive) algorithms. Global approaches have the advantage that they avoid obstacles with globally optimal paths previously calculated. Usually these methods employ environment information and process it to make the final movement of the robot. However, they cannot effectively deal with cases where aleatory changes in the obstacle movement or in the environment occur. An overview of this class of methods can be found in Latombe's work [28]. On the other hand, local obstacle avoidance methods [29], [30] and [31] treat the problem in a reactive way, using local information to resolve the problem of navigation. Thus they are able to adapt the movement of the robot to aleatory changes in the environment.

The current navigation system implemented for the Nicole platform is an obstacle avoiding algorithm based in potential fields methods. During the past few years, potential field methods (PFM) for obstacle avoidance applications have gained popularity among researchers in the field of mobile robots. The idea behind all these methods is the existence of imaginary forces around the robot, attraction forces to represent the target of the final movement, and repulsive forces to show the different obstacles around the robot [29][32]. This information is presented in an unified system and process to obtain the next direction in a specific instant of time.

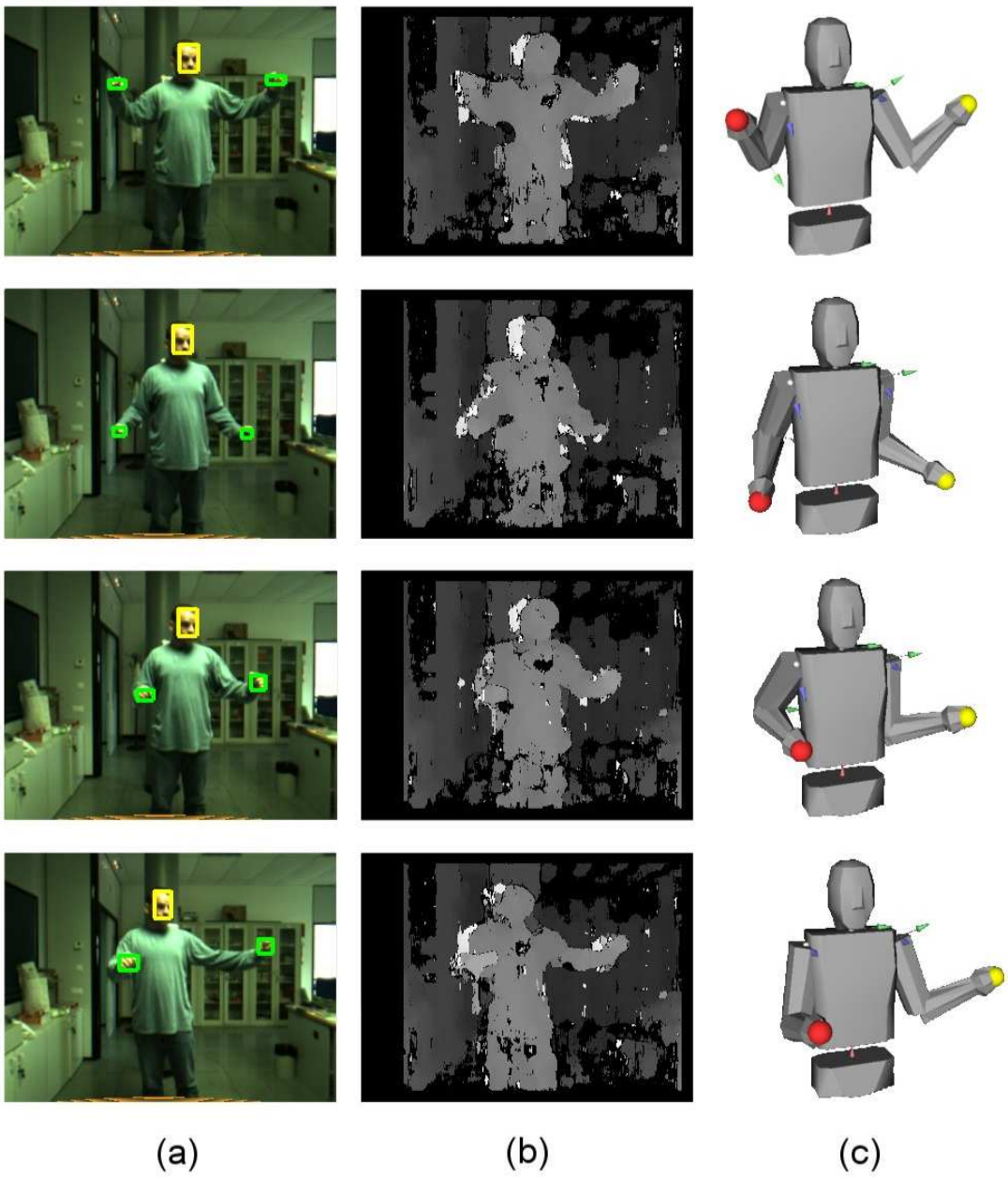


Figure 33: Human Motion Capture system: a) Left image of the stereo pair with head (yellow) and hands (green) regions marked; b) Disparity map; and c) 3D model showing generated pose.

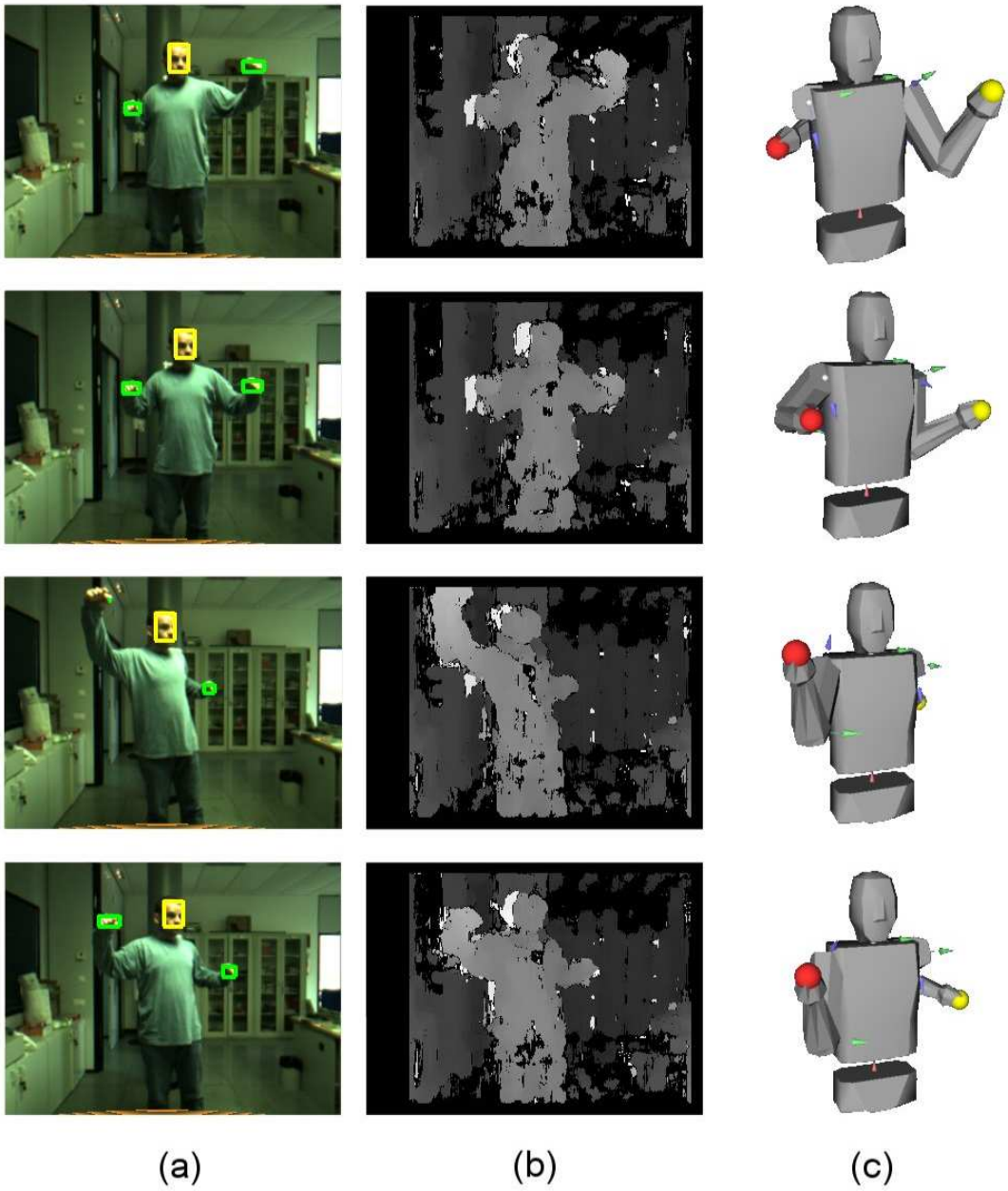


Figure 34: Human Motion Capture system: a) Left image of the stereo pair with head (yellow) and hands (green) regions marked; b) Disparity map; and c) 3D model showing generated pose.

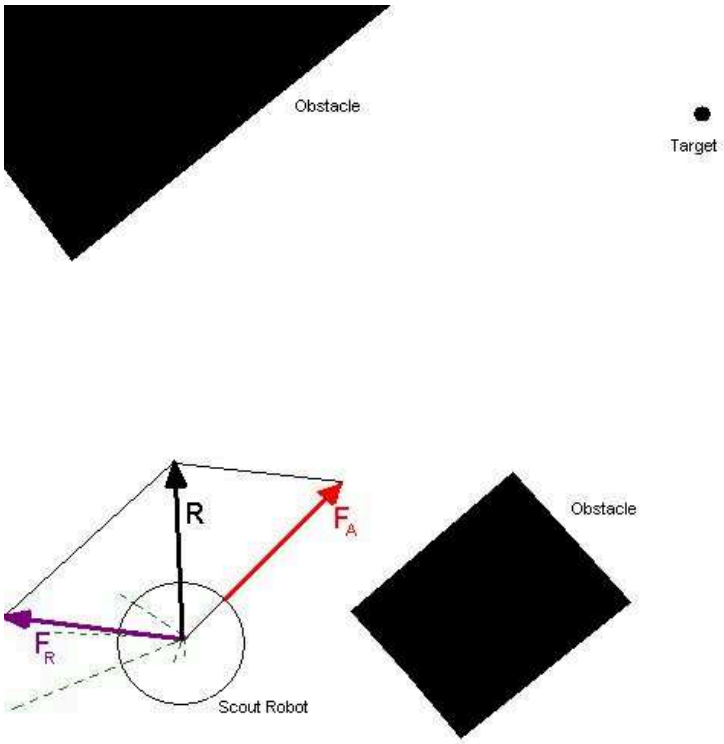


Figure 35: Potential fields concepts. Different forces involved.

4.8.2 Potential Fields Algorithm Description

Potential field method has been implemented on Scout Nomad mobile robot with sensory data. The Sensus 200 is a ring of 16 Polaroid 6500 sonar ranging modules. The Polaroid 6500 is an acoustic range finding device that has been widely used in the mobile robotics community. It can measure distances from 6 inches to 35 feet, with a typical absolute accuracy of 1 percent over the entire range.

In our navigation system the environment information is continuously acquired using this sonar ring. Thus the robot gets knowledge about different obstacles around itself. The potential fields algorithm considers measures of the sonar as repulsive forces F_{Ri}. This repulsive force is characterized as a force vector whose start point is the center of the robot and its direction corresponds to the negative of the angle to the obstacle. The length of this vector is related to the pose of the robot (we can consider that nearest

obstacles provokes more repulsion than far obstacles). Finally, the resultant repulsive force F_R is consequence of the sum of each F_{Ri} as:

$$F_R = \sum_{i=0}^{i=16} F_R i \tag{23}$$

On the other hand, the target of the robot generates an attractive force which can also be approximated as a vector F_A. This vector has a start point in the center of the robot, its direction corresponds to the angle to the target, and its length is a fixed value defined in the algorithm. Finally, These two forces, F_R and F_A are summed and the resultant force R determines the subsequent direction and speed of travel.

$$R = F_R + F_A \tag{24}$$

Fig. 35 presents potential field concepts. The target generates an attraction force F_A to this point (red line) while obstacles generate repulsive forces (discontinuous green line in the image). The sum of all these forces is F_R, violet in the figure, the repulsion force that face to F_A. The result of the sum is the final direction of the robot.

4.9 Gesture perception

In this Section, we present a system that extracts the gesture-features of a human actor from a series of images taken by a single camera. Section 4.9.1 presents the theory behind the gesture recognition with a probabilistic model using a Bayesian framework. In Section 4.9.2 we address issues of the implementation of the gesture recognition system including the learning process. In Section 4.9.3 we show some experiments related with the performance of the gesture recognition and some issues related to the learned tables. Section 4.9.4 closes with conclusions.

4.9.1 Means of Interaction - Gesture Libraries

The communication from the human to the robot will be based on hand movements conveying useful information, i.e. hand gestures. This raises two questions to be answered: 1) What makes a movement to appear as a gesture and 2) What is a useful set of gestures? To tackle the first question we start with a concept proposed for human motion analysis. As a gesture is created

Recognized Gesture	Interpretation	Action
Draw Circle	Turn 360	Rotation
Saggital Waving	Come closer	Move forward
Horizontal Waving Left side	Step aside (left)	Move right
Horizontal Waving Right side	Step aside (right)	Move left
Waving Bye-Bye	Ignore last gesture Stop Interaction	Wait Switch off system
Pointing gesture	Change godfather Acknowledgement	Proceed to next point Perform Action

Table 7: Gesture-Action Mapping

Figure 36: Gesture phases: a) Pre-Stroke b) Stroke c) Post-Stroke

by motion we need to find an appropriate description for the spatio-temporal behavior. We develop 'atomic' segments of gestures which we can relate to our observation sequence.

A suitable model is to divide the gesture into three phases [33]: 1. *Pre-stroke* (preparation), 2. *Stroke* and 3. *Post-stroke* (retraction). Figure 36 a) - c) shows an example for a deictic gesture (i.e. pointing gesture). Gesture recognition systems have often adopted this temporal composition [34, 35]. In [36] the phases are called 'phonemes' following the terms used in phonology to describe the principal sounds in human languages. The second question may be rephrased thus: What kind of knowledge about the world do I need to provide to the robot? The set of gestures need to be rich enough to trigger a certain variety of actions and the gestures must be intuitively and effortlessly performed by the human.

The 'Nicole' dictionary maps a set of gestures into actions to be executed

(see table 7). In this table we distinguish between a recognized gesture (e.g. Waving Bye-Bye) and its interpretation i.e. the meaning it represents. The interpretation depends on the context given by a higher level module called action-planner. A simple example is a sequence of two gestures. If the first gesture is a "*Waving Bye-Bye*" and the second is "*Pointing*", the system will assume a command to switch off the system followed by its acknowledgement. If the sequence would be the other way round, the system would assume a command to proceed to the next waypoint which was canceled.

The set of gestures has been organized into three categories: 1) *Control Gestures*, 2) *Pointing Gestures* and 3) *Social Gestures*. Category 1 gestures are used to control movements and audio output like 'move to the left'. Such sets have already been used in the past to control actuated mechanisms [37]. Category 2 are gestures that are meant to shift Nicole's focus of attention to a certain direction (deictic gestures). Pointing gestures have already been used in the past to search and find objects in an image [38]. The last category covers useful social gestures like 'Waving Bye-Bye'.

4.9.2 Theory

Why using the Bayesian approach? Our goal is to design a probabilistic model using a Bayesian framework to anticipate the gesture given the observed features. The Bayesian framework can offer combinations of the whole family of probabilistic tools like Hidden Markov Models (HMMs), Kalman Filters and Particle Filters and their various modifications. Though, the Bayesian framework can be used for all kind of system modeling (e.g. navigation, speech recognition, etc.) they are specially suited for cognitive processes. Research on the human brain and in its computations for perception and action report that Bayesian methods have proven successful in building computational theories for perception and sensorimotor control [39]. The process of prediction and update represents an intrinsic implementation of the mental concept of anticipation. In general, modeling offers the opportunity to reach a modest dimensionality of the parameter space that describes the human motion. Bayesian models in particular also maintain an intuitive approach which can also be understood by non-engineers [40]. Furthermore these methods have already proven their usability in gesture recognition [34, 35].

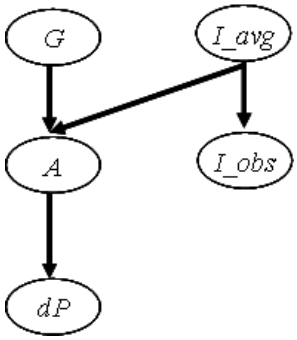


Figure 37: Bayesian Net for the gesture model.

Bayesian Inference The solution using the Bayesian approach assumes that for a given frame i from all possible frames I and a given gesture g from all possible gestures G the probability that the atom has value a, which is $P(a|g,i)$ can be determined. More general, we can express the probability distribution for all possible values of atom A given all possible gestures G and frames I with $\mathbf{P}(A|G,I)$.

Knowing the conditional probability $\mathbf{P}(A|G,I)$ together with the prior probabilities for the gestures $\mathbf{P}(G)$ we are able to apply Bayes rule and compute the probability distribution for the gestures G given the frame I and the atom A:

$$\mathbf{P}(G|I,A) = \mathbf{P}(G)\mathbf{P}(A|G,I) \tag{25}$$

The term $\mathbf{P}(A|G,I)$ is represented as a big look-up table while $P(a|g,i)$ is the entry of a certain cell that answers the query "What is the probability that atom has value a if gestures has value g and frame has value i?".

The next step is to compute the probability that a certain gesture has caused the whole sequence of atoms. If we assume that the observed atoms are independently and identically distributed (i.i.d.) we are able to express the joint probability by the product of the probabilities for each frame.

$$P(a_{1:n}|g,i_{1:n}) = \prod_j P(a_j|g,i_j) \tag{26}$$

Where $a_{1:n}$ represents the sequence of n observed values for atom and g a certain gesture from all gestures G. The jth frame of a sequence of n frames is represented by i_j. Equation 26 is also known as the *likelihood computation.*

At this point it might be useful to point out that the different ways to model the world in a Bayesian way usually give names to these approaches.

As, in our case we assume that any value of the random variable A is similar likely to appear we may associate the name 'Naive Bayes' to the approach. It distinguishes itself from approaches like 'Hidden Markov Models' or 'Kalman Filter' that we assume that the current value a_t was not influenced by the previous value a_{t-1}. This 'naive' way to model the world is ,surprisingly, also the most successful.

We are now able to express the probability of a gesture g that might have caused the observed sequence of atoms $a_{1:n}$ by plugging equation 26 into equation 25. We formulate this in a recursive way. Assuming that each frame a new observed atom arrives we can state and expressing the real-time behavior by using the index t:

$$\mathbf{P}(G_{t+1}|i_{1:t+1}, a_{1:t+1}) = \mathbf{P}(G_t)\mathbf{P}(a_{t+1}|G, i_{t+1}) \tag{27}$$

We see that the probability distribution of the gestures G at time $t+1$ knowing the observed atoms a until $t+1$ is equal to the probability distribution of G at time t times the probabilities of the current observed atom given the gestures G and frame i at $t+1$. The probability distribution of G for $t = 0$ is the prior discussed with equation 25. We will later see that, as more observed atoms arrive, the probability distribution of the gestures will converge to the correct gesture even if the prior was wrong. This will happen for any fixed prior, as long as it does not rule out the correct gesture by assigning zero probability to it.

Due to the number of repetitions and the pace of the performer we can not assume a fixed number of atoms per gesture. From experiments we can estimate the mean and variance of a average gesture performance in terms of a Gaussian distribution $N(i\_obs, \sigma)$. With this we can express the probability that an observed frame i_obs maps to an average frame iavg P(iobs — iavg).

$$P(i\_obs|i\_avg) = N(i\_obs; \sigma) \tag{28}$$

We are now able to formulate our Bayesian model by plugging equation 28 into equation 27.

$$\begin{aligned} &\mathbf{P}(G_{t+1}|i_{1:t+1}, a_{1:t+1}) \\ &\quad = \mathbf{P}(G_t)P(i\_obs_{t+1}|i\_avg_{t+1})\mathbf{P}(a_{t+1}|G, i_{t+1}) \end{aligned} \tag{29}$$

We can likewise express our model in a *Bayesian Net* shown in fig. 37. It shows the dependencies of the above mentioned variables including the

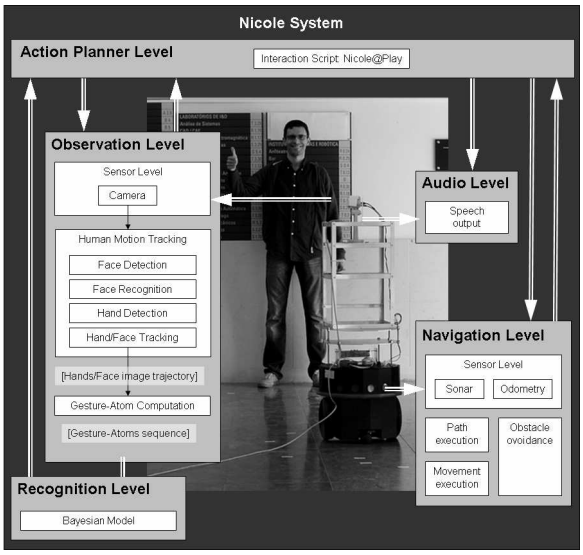


Figure 38: Architecture of the GP-System.

displacement dP from the previous section. The rule for classification is based on the highest probability value being above a certain threshold while an unknown gesture, i.e. an unknown sequence of atoms produces more than one gesture-hypothesizes with a significant probability.

4.9.3 Implementation

In [41] the complete system architecture of the guide robot "Nicole" was presented as well as the architecture of the GP-System. Figure 38 shows the main parts of the system architecture omitting the *Action Planner* which controls the sequential execution of the tasks inside the interaction-scenario.

The system is divided into three levels starting inside the *Observation Level* with the visual sensor dealing with image capture. The image data is used by the *Human (Motion) Tracking* module to perform face detection, face recognition, skin-color detection and object tracking and has been described in [41]. We use a face detection module based on haar-like features as described in [42] and a face recognition based on eigen-objects and PCA [43]. For skin detection and segmentation we use the CAMshift algorithm presented in [44]. From the resulting trajectories we calculate the relative displacement between each frame and the absolute displacement from the initial position. The latter triggers the starting and end of the gesture. The

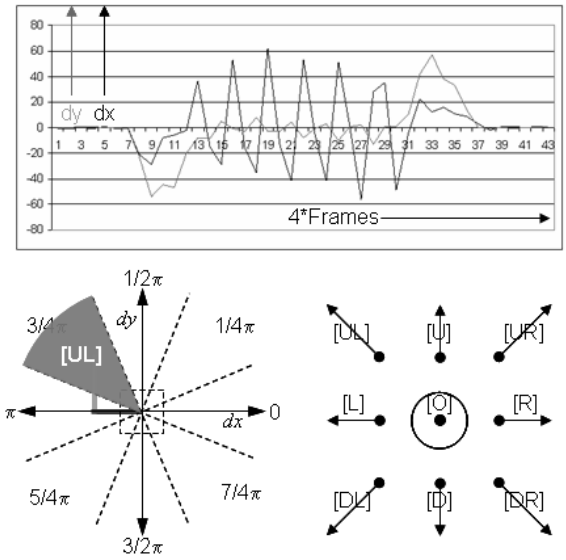


Figure 39: Computation of atoms from the displacement signal.

former undergoes a further discretization while computing principal motion motion vector referred to as *Gesture Atom*. This approach is reminiscent of the concept of phonemes and words from speech recognition as the sequence of atoms will form a specific gesture. As seen in fig. 39 we calculate the displacement dx, dy every 4th frame (roughly 4 times per second). The atom is given by the sector of the displacement vector. In an extended view, at the *Observation Level*, the *HandFace Tracking* module is looking to the displacement vector of the hands waiting for a trigger to start and stop collecting *Atoms*. The features extracted in the *Observation Level* will be used by the Recognition Level.

The Recognition Module is proceeded by a Learning process of the known gestures. The Learning Process is accomplished based on set of images previously stored in a Database of Gestures. With different performers we store at least 6 sequences for each gesture we later try to recognize. Both the gestures and the sequences used in the Learning Process are defined in text files so that each time the program executes the Learning Process looks these files to find out where from learn the data. In this way we are able to define what kind of gesture we what to recognize but also the data used to learn a gesture.

The Learning Process addresses to create a conditional probability table

expressing $P(A|GI_{avg})$. The computation of the Learning Process simply counts the number of atoms that occurs in a frame for a gesture. The size of the table varies according to the number of frames in each sequence, by taking the maximal length of the set of sequences for a gesture. Assuming that we are able to recognize 6 different gestures which can be performed with one or two hands, we get a table size $18Atoms$ x $Maxframes$ x $6Gestures$. Once this process is done online this table is ready to be used by the Recognition Module.

The module of the Recognition Level is implemented using a special library - ProBT, from ProBayes company, that provides probabilistic computation tools. Using the ProBT library we start by building a set of variables needed to compute the Bayesian network. Gesture type, atoms for left and right and frame are basic variables for our model. In our model it is possible to have 9 different atoms, see fig. 39 and 6 different gestures. The frame set is defined varying from 1 to 50. The probabilities of a gesture and a frame are defined as uniform distributions. To avoid the uncertainty of the frame-length of a gesture we define a variable, observed frame, which can vary between 1 and $frames*1.25$. The probability of an observed frame is defined as a bell-shape distribution with mean given by the frame and variance of 1.5. Our focus is in the gesture type, thus, we built a network using the gesture, the frame, the observed frame and the left and right atoms, putting then a question for the gesture type knowing both left and right atoms and also the observed frame. In each iteration the probability of the gestures is computed and the result is next used to produce a convergent result to a gesture type. As referred before the Recognition module stops either using a threshold for the probability of a gesture or a signal from HandFace Tracking module. At this moment the Recognition system cleans up and waits for instructions for a new computation.

The module will recognize a gesture from the known vocabulary. The action planner will interpret the perceived gesture depending on the context. The following *Interaction Level* will initiate actions like speech output or motion commands according to the interpreted gesture.

4.9.4 Experiments

In our experiments human actors were performing a set of six gestures see 15 times each. The sequences taken by the camera are stored in a database for future replications. We compute the image trajectories of hands and head,

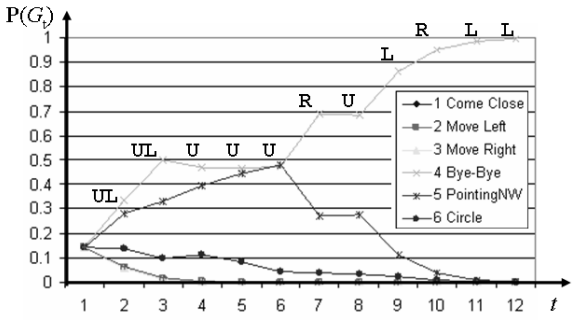


Figure 40: Probability evolution for a Bye-Bye gesture input.

the sequence of gesture atoms and the probability values of the gestures. To provide a ground truth for our image tracker we also collect data from a magnetic tracker. The contents of this database are publicly available at the project's web page (http://paloma.isr.uc.pt/nicole/).

Figure 40 illustrates how the gesture-hypothesizes, evolve as new evidences (atoms) arrive taken from the performance of a Bye-Bye gesture. After twelve frames the probabilities have converged to the correct gesture-hypothesis (light-blue). After four frames the probabilities of the two hand gesture-hypothesis have reached nearly zero. (blue, pink, move left, move right). Until the sixth frame the probabilities of both head-level gestures grows (light-blue and purple) which indicates the pre-stroke phase. Conversely the probability of the belly-level gesture (Circle) drops slowly towards zero. After the sixth frame the oscillating left-right movement (and its associated atoms) makes the probability of the Bye-Bye-gesture hypothesis rise and the Pointing-NW-gesture hypothesis drop. The results for the other gestures are similar.

4.9.5 Bayesian Learning

As both, the gestures and the frame index are discrete values we can express $P(A|GI_avg)$ in form of a conditional probability table. The probabilities can be learned from training data using a certain number of atom-sequences for each gesture. A simple approach is the one known as Histogram-learning. It counts the number of different atom-values that appear for a gestures along the frames. To overcome the problem of assigning zero probabilities to events that have not yet been observed an enhanced version often uses learning of a

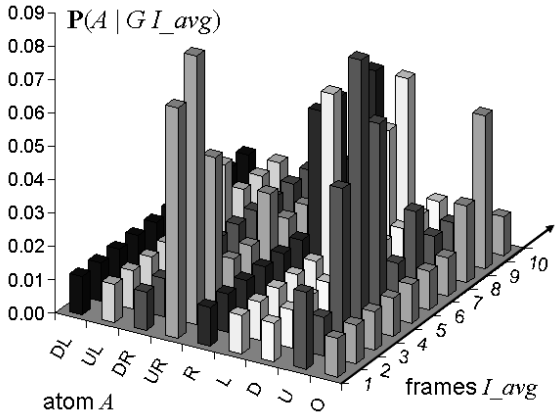


Figure 41: Learned Table $P(A|GI\_avg)$ for gesture 'Bye-Bye'.

family of Laplace-distributions. Currently we are using a table that is of size 18 x 31 x 6, that is 18 discrete values for the atom (9 for each hand), 31 frames and 6 gestures. Figure 41 shows a fraction of the table which is the 9 atoms of the right hand for the first 11 frames and the Bye-Bye gesture. It represents the 'fingerprint' of the gesture prototype for waving Bye-Bye. Knowing the gesture we assume this sequence of atoms to be extracted. More precisely we assume this sequence of distributions of the random variable atom to extracted. Of course, like any matrix gained like e.g. PCA or ICA it tries to show the components that represent. an observation best. The big difference is, that it keeps an intuitive way to store the information. If we take a look at fig. 41 we can see, that during the first frames the most likely atom to be expected is the one that goes Up-Right (UR). This coincides with our intuition, that while we are starting to perform a Bye-Bye gesture with the left hand we tend to move up and to the left to gain space to perform the gesture. At this point it might be useful to explain the we have named the actions according to the performers point of view (i.e. pointing to his left we refer as pointing East) while the atoms are named due to the observes point of view (i.e. pointing to the left will generate right atoms). Several useful conclusions can be drawn directly from the learned tables, mainly related from what we saw in 4.9.1 concerning the three phases of a gesture. Related with this we are also able to infer the position of the the hands (waist, belly or head).

Looking at fig. 42 we can discuss how the different gestures can be dis-

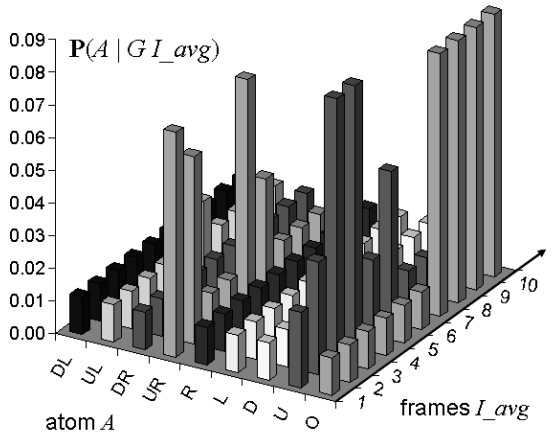


Figure 42: Learned Table $P(A|GI\_avg)$ for gesture 'Pointing NW'.

tinguished intuitively. The gesture of pointing to the northwest (forward and to the left) is usually performed by the left hand. It shows that during the first frames the most probable value for an atom is, again Up-Right (UR). This makes sense as during the Pre-stroke phase both gestures need to reach the space on the left side of the head. The distinction can be done only after the seventh frame. The pointing gestures will produce mainly zero-motion atoms (O) while the waving gestures will have a roughly equal distribution along the line of oscillation (e.g. left-right).

The discussion on the learned probability tables could also be turned toward the distinctions of different people (e.g. person A performs the gesture like this, person B like that) or in the direction of emotional content (e.g. a person at ease performs the gesture like this, a person who is excited like that). In any case, keeping the 'fingerprints' intuitively will make the evaluation more convincing.

4.9.6 Discussion and results

This work demonstrates that Bayesian approaches provide a robust and reliable way to classify gestures in real-time. Using naive Bayesian classification we are able to anticipate a gesture from its beginning and can take decisions long before the performance has ended.

We have show that the Bayesian way in which the system stores learned gestures is intuitive and provides the possibility to draw conclusions directly

from the look-up tables.

Performance measures were defined to evaluate the skill of anticipation inside a the context of human-robot interaction.

The whole system is implemented on a mobile platform which provides the possibility to test social interaction in various scenarios.

We are currently working on a system that maps the anticipated gestures to an avatar using a similar Bayesian approach presented here for recognition. We aiming towards a social platform where the impact of imitation between a human and a machine can be observed.

We hope that the output of the imitation model can also serve as a top-down approach to help in the tracking process.

Furthermore we want to tackle the problem of empathy by extracting features that enables use the recognize the intention of a performance. Again, searching for a solution using a Bayesian framework.

5 Meetings and public demonstrations

5.1 Meetings and researcher exchanges

The kick-off meeting of the VISOR project took place in Coimbra on September 19th. In this meeting, the VISOR objectives were clearly defined and deliverables, milestones and exchange of researchers within VISOR were discussed. A brief technical session took also place with presentations by P. Menezes, R. Marfil, J.P. Bandera and J. Rett. The main theoretical aspects of VISOR project were defined in these presentations.

The second VISOR meeting took place on December 2nd, 2005 in Mlaga. The main goals of the meeting were to finish the first VISOR deliverable and to discuss the organization of a special workshop on Visual based Human-Robot Interaction, which was held in conjunction with EUROS 2006 in Palermo (Italy). The final programme of this workshop included eight papers, and it was a good scenario to discuss different aspects (grasping, human motion capture, timing of visual and spoken input, tracking...) of the visual based human-robot interaction. R. Marfil, L. Molina-Tanco and P. Menezes presented their work in this workshop.

The third VISOR meeting took also place on March 18th, 2006 in Palermo. Future milestones and researcher exchanges were defined. A first stay was made by J. Rett, from the ISR group, at the University of Málaga in April

2006. In this stay, the VISOR project demos were defined. From May to July 2006, Juan Pedro Bandera and Pedro Núñez, from the ISIS group, were visitors at Coimbra University. This stay permitted the major part of the VISOR team to be grouped at Coimbra for three months.

5.2 Coimbra trials

At the end of this stay, on July 2006, the fourth VISOR meeting and the first demonstration of the VISOR project took place on Coimbra. The technical aspects of the human motion capture and human-robot interaction applications were initially presented. The public at Coimbra University then were able to play with Nicole, the social robot from ISR-Coimbra (Fig. 43) or to look at how their movements were captured and imitated by a synthetic model.

5.3 Málaga trials

The results of the previous demo at Málaga highlighted some technical issues which were tackled over the summer, when a final integration effort was made. The last VISOR meeting took place in Málaga on September 11th. In an open-door day, the Coimbra experience was repeated at Málaga (Fig. 44).

6 Conclusions reached

This Topical Research Study has investigated the scientific challenges to the development of a visual perception system for a socially interactive robot. These are the main technical conclusions reached by the partners involved in the project.

The combination of state of the art techniques in **face detection** and **skin color segmentation** constitutes a powerful tool for a socially interactive robot. Face detection is without any doubt a must-have skill which allows the social robot to become aware of the presence of a human in order to trigger social behaviour. Of course face detection techniques allow a robot to detect a 'social human', i.e. one that is willing to interact with the robot and are thus unsuited to security applications.

In this study, a new approach to face detection has been introduced that runs at video frame rate. This is achieved by combining skin colour segmen-

Figure 43: Nicole in front of a godfather at University of Coimbra (July 2006).

Figure 44: Human motion capture system demonstrated at University of Mlaga (September 2006).

tation with detection based on Haar-like features (Section 4.4). However, computational resources in the social robot must be shared with other tasks, and thus it is recommended that a face detection behaviour is triggered only when needed. Such is the choice of the architecture of Section 4.7: once a face is detected, colour tracking suffices to follow the human. If the face detection is only fired when required, this means that it can be done independently of skin colour, and at many scales, which allows for detection of humans independently of their race, size, gender or age, with varying lighting conditions, indoors and outdoors, and at different distances to the robot.

By integrating knowledge on how humans move, and with the support of fast, robust **colour tracking**, and **depth estimation** through stereo triangulation, this project has demonstrated the feasibility of visual real-time human motion capture which will endow an anthropomorphic robot with an essential social skill: the ability to imitate human motion, using only its own sensors (Section 4.7). Real-time, visual human motion capture will allow the social robot to give real-time feedback to the human as to how the demonstrated motion is being understood.

Stereo triangulation requires a calibrated camera pair. If stereo vision is not available, monocular vision can still be used by the social robot for **gesture recognition**. This work has investigated Bayesian approaches to gesture recognition. We have demonstrated that they provide a robust and reliable way to classify gestures in real-time. Using naive Bayesian classification we are able to anticipate a gesture from its beginning and can take decisions long before the performance has ended. The developed architecture allows the implementation of the gesture recognition system on a mobile platform which provides the possibility to test social interaction in various scenarios.

7 Impact and disseminations of results

7.1 Public deliverables

- R. Marfil and J. Rett, "Skin Colour Detection, Face Detection and Face Recognition", *Visual Perception System for a Social Robot (VISOR) Project*, Deliverable 1, Institute of Systems and Robotics, Coimbra, Portugal, December 2005.

- R. Marfil, "Attentional Mechanism", *Visual Perception System for a Social Robot (VISOR) Project*, Deliverable 2, Grupo de Ingeniería de Sistemas Integrados, Málaga, Spain, March 2006.

- J.P. Bandera, "Human motion capture", *Visual Perception System for a Social Robot (VISOR) Project*, Deliverable 3, Grupo de Ingeniería de Sistemas Integrados, Málaga, Spain, June 2006.

7.2 Publications

- R. Marfil, L. Molina-Tanco, A. Bandera, J.A. Rodríguez and F. Sandoval. Pyramid segmentation algorithms revisited, *Pattern Recognition* 39, pp. 1430-1451, 2006.

- R. Marfil, L. Molina-Tanco, J.A. Rodríguez and F. Sandoval. Real-time object tracking using Bounded Irregular Pyramids, (submitted the third revised version to *Pattern Recognition Letters* to satisfy minor revisions).

- J. P. Bandera, R. Marfil, L. Molina-Tanco, J. A. Rodríguez, A. Bandera and F. Sandoval. Robot learning of upper-body human motion by active imitation, *2006 IEEE RAS International Conference on Humanoid Robots* (accepted).

- J. P. Bandera, L. Molina-Tanco, R. Marfil, A. Bandera and F. Sandoval. An Active Vision System for a Social Robot. *(journal paper in preparation).*

- R. Marfil, R. Vázquez-Martín, L. Molina-Tanco, A. Bandera and F. Sandoval. Fast Attentional Mechanism for a Social Robot, *Workshop on Visual based Human-Robot Interaction (held in conjunction with EUROS'06)*, Palermo-Italy, 2006.

- L. Molina-Tanco, J.P. Bandera, J.A. Rodríguez, R. Marfil, A. Bandera and F. Sandoval. A Grid-based Approach to the Body Correspondence Problem in Robot Learning by Imitation. *Workshop on Visual based Human-Robot Interaction (held in conjunction with EUROS'06)*, Palermo-Italy, 2006.

- P. Menezes, F. Lerasle and J. Dias. Visual Tracking Based Modalities Dedicated to a Robot Companion, *Workshop on Visual based Human-Robot Interaction (held in conjunction with EUROS'06)*, Palermo-Italy, 2006.

- Rett, J., Dias, J. Visual based human motion analysis: Mapping gestures using a puppet model. *Proceedings of EPIA 05, Lecture Notes in AI* Springer Verlag, Berlin, 2005.

- Rett, J., Dias, J. Gesture Recognition Using a Marionette Model and Dynamic Bayesian Networks (DBNs) *Proceedings of ICIAR 2006, Lecture Notes in CS 4142* Springer Verlag, Berlin, 2006

- Rett, J., Dias, J. Gesture Recognition based on Visual-Inertial Data - Registering Gravity in the Gesture Plane. To appear in: *Proceedings of the Colloquium of Automation*, Salzhausen 2005/2006, 2006 - (Best Paper Award)

- Rett, J., Dias, J. Bayesian Learning and Gesture Anticipation in the context of Human-Robot Interaction. Submitted to ICRA'07.

7.3 PhD projects

The VISOR project has partially granted a number of PhD students both at Coimbra University and University of Málaga. What follows is a list of their names and the subjects of their research in the area of social robotics.

7.3.1 Awarded

- Rebeca Marfil. *Tracking objects with the Bounded Irregular Pyramid.*, Ph.D. Dissertation, Dpto. Tecnología Electrónica, Universidad de Málaga. May 2006.

  The main contributions of this Thesis are:

  - The implementation and detailed analysis of a new pyramidal structure for image processing: the Bounded Irregular Pyramid (BIP). The key idea of this pyramid is to combine the advantages of regular and irregular pyramids within the same structure. To do that regular and irregular data structures as well as regular and irregular decimation processes are mixed in a novel way to build the BIP. This pyramid allows to process images ten times quicker than the existing irregular pyramids with similar accuracy. This reduction of the computational time makes it possible to use the BIP in real-time applications.
  - The development of a new template-based target representation scheme using the Bounded Irregular Pyramid. This template combines colour and spatial information. The way in which this template is updated allows to include information of previous templates in order to avoid tracking errors due to appearance changes of the object or occlusions. Besides, because of the structure of the BIP, the proposed target representation approach allows to take into account neighbourhood information of each pixel of the template in the tracking process.
  - The implementation of a tracking algorithm based on template matching. This algorithm takes advantage of the hierarchical structure of the template representation to perform the template matching in a hierarchical way. This approach makes possible to simultaneously track several objects without a high increase of the computational cost. A version of the tracking algorithm developed

in this Thesis is employed in the attention mechanism described in Section 4.6, and subsequently in the human motion capture system described in Section 4.7.

7.3.2 In progress

- Pedro Núñez.

  The research is focused at an human-robot interaction system for indoor environment. The robotic platform includes a navigation system based in a Simultaneous Localisation and Map building (SLAM) with landmarks extracted with a laser range sensor. After the robot is localized in the world, it will be able to interact with humans using visual (face and gesture recognition) and audio information.

- Ricardo Vázquez-Martín.

  This research is focused on simultaneous localization and map building (SLAM) problems for mobile robots in structured (indoor) and unstructured (outdoor) environments. The implementation of the SLAM algorithm is based on the extended Kalman filter (EKF-SLAM) and use several techniques and sensors to extract landmarks from the environment. Laser scans are used in indoor like environments and an attention mechanism based on active vision, using a stereo vision system, is used for indoor and outdoor environments.

- Juan Pedro Bandera.

  This thesis focus on the development of a vision-based learning by imitation system. The main objective is to make a humanoid robot recognize human gestures in typical social interaction scenarios. This involves the implementation of the human motion capture system described in this report, the analysis and parameterization of perceived movements and the translation of these movements from human to humanoid.

- Joerg Rett.

  This thesis focus in robot vision for human machine interaction. The development is focused on systems to be implemented on autonomous platforms, mainly in the interface between humans and robots (e.g. mobile robots). To make such a system reliable several input modalities

need to be integrated. The author is currently creating an interface that translates and elevates occurrences in the image-based level to higher levels of perception.

8 Self assesment

The project achieved different tangible goals, which are also presented on the project's web page (http://www.grupoisis.uma.es/visor/VIsual System for a SOcial Robot-Home.htm).

The research inside the project resulted in one journal publication, six publications at conferences, one presentation in a colloquium for later publication , two journal submissions, and one submission to a conference (See Section 7). Four public deliverables were produced including this final report, which are accessible through the project's web page. o Two prototypes were developed. The Social Robot Nicole, a multipurpose platform to investigate social interaction between humans and robots, and a real-time, vision-based human motion capture system that can estimate upper-body motion from a camera pair. The prototypes have been presented to the public during two demo events. The first event took place on Monday 10.July 2006 at the University of Coimbra, Polo 2 in Coimbra, Portugal, the second event was hold on Monday 11.September 2006 at the Technology Park in Mlaga, Spain. The events have produced additional material like movies and posters that are accessible through the project's web page (http://www.grupoisis.uma.es/visor/VIsual System for a SOcial Robot-Home.htm).

One workshop was organized at an international conference (EUROS-2006) to foster the work in the field of Human-Robot Interaction. The International Workshop on Vision Based Human-Robot Interaction was held inside the EUROS-2006 conference on Saturday 18. March 2006 in Palermo, Italy and its content can be accessed at http://paloma.isr.uc.pt/ hri06/, the event's web page.

References

[1] A. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[2] R. Marfil, A. Bandera, J. Rodríguez, and F. Sandoval, "Real-time template-based tracking of non-rigid objects using bounded irregular pyramids," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, vol. 1, pp. 301–306, 2004.

[3] J. P. Bandera, R. Marfil, L. Molina-Tanco, A. Bandera, and F. Sandoval, "Model-based pose estimator for real-time human-robot interaction," *Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

[4] L. Molina-Tanco, J. P. Bandera, R. Marfil, and F. Sandoval, "Real-time human motion analisys for human-robot interaction," *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1808–1813, 2005.

[5] G. Backer and B. Mertsching, "Two selection stages provide efficient object-based attentional control for dynamic vision," *Proc. Int. Workshop Attention and Performance in Computer Vision*, pp. 9–16, 2003.

[6] J. Terrillon and S. Akamatsu, "Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images," *Proc. 12th Conf. on Vision Interface*, vol. 2, pp. 180–187, 1999.

[7] I. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: A five year experiment," in *IROS 2003*, 2003.

[8] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The interactive museum tour-guide robot," in *AAAI/IAAI*, 1998, pp. 11–18. [Online]. Available: citeseer.ist.psu.edu/burgard98interactive.html

[9] R. Siegwart and et al., "Robox at expo.02: A large-scale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42 No. 3-4, pp. 203–222, 2003. [Online]. Avail-

able: http://www.sciencedirect.com/science/article/B6V16-47X20PK-6/2/c0a0f251fac05ba67b7a0fcfd34b17b

[10] J. C. Terrillon, M. David, and S. Akamatsu, "Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments," *Proc. Int. Conf. Face and Gesture Recognition*, pp. 112–117, 1998.

[11] J. C. Terrillon and S. Akamatsu, "Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images," *Proc. 12th Conf. on Vision Interface*, vol. 2, pp. 180–187, 1999.

[12] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detcting faces in images: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 34–58, 2002.

[13] P. Viola and M. Jones, "Robust real-time face detection," *Int. Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[14] Y. Freund and R. Schapire, "A decision-theoric generalization of on-line learning an application to boosting," *Proc. European Conf. Computational Learning Theory*, pp. 119–139, 1995.

[15] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Int. Conf. Computer Vision*, pp. 555–562, 1998.

[16] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103–108, 1990.

[17] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[18] *OpenCV Reference Manual*, Intel.

[19] L. Itti, "Real-time high-performance attention focusing in outdoors color video streams," *Proc. SPIE Human Vision and Electronic Imaging (HVEI'02)*, pp. 235–243, 2002.

[20] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 161–169, 2001.

[21] R. Marfil, "Attentional mechanism," Visual Perception System for a Social Robot (VISOR) Project, Deliverable 2, Grupo de Ingenieria de Sistemas Integrados, University of Malaga, Spain, Tech. Rep., 2006.

[22] E. Sauser and A. Billard, "View sensitive cells as a neural basis for the representation of others in a self-centered frame of reference," *Proceedings of the Third International Symposium on Imitation in Animals and Artifacts, Hatfield-UK*, pp. 119–127, April 2005.

[23] C. R. Wren and A. P. Pentland, "Dynaman: Recursive modeling of human motion," Technical Report TR-451, MIT, Tech. Rep., 1999.

[24] Y. Nakamura and K. Yamane, "Dynamics computation of structure-varying kinematic chains and its application to human figures," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 2, pp. 124–134, 2000.

[25] J. R. Mitchelson, "Multiple-camera studio methods for automated measurement of human motion," Ph.D. dissertation, CVSSP, School of Electronics and Physical Sciences, Univ. of Surrey, UK, 2003.

[26] J. J. Craig, *Introduction to Robotics*. Addison-Wesley, 1986.

[27] S. Gottschalk, M. C. Lin, and D. Manocha, "Obb-tree: A hierarchical structure for rapid interference detection," Technical Report TR96-013, Department of Computer Science, University of N. Carolina, Tech. Rep., 1996.

[28] J. C. Latombe, "Robot motion planning," *Kluwer Academic Publishers*, 1991.

[29] O. Khatib, "Real-time obstacle avoidance for robot manipulator and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[30] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[31] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proc. of 1996 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, Minneapolis, USA, April 1996, pp. 3375–3382.

[32] J. R. Andrews and N. Hogan, *Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator*, ASME, Boston, 1983, pp. 243–251.

[33] N. Rossini, "The analysis of gesture: Establishing a set of parameters," in *Gesture-based Communication in Human-Computer Interaction, LNAI 2915, Springer Verlag*, 2003, pp. 124–131.

[34] T. Starner, "Visual recognition of american sign language using hidden markov models," Master's thesis, MIT, Feb 1995.

[35] V. I. Pavlovic, "Dynamic bayesian networks for information fusion with applications to human-computer interfaces," Ph.D. dissertation, Graduate College of the University of Illinois, 1999.

[36] S. Kettebekov, M. Yeasin, and R. Sharma, "Prosody based co-analysis for continuous recognition of coverbal gestures," in *International Conference on Multimodal Interfaces (ICMI'02)*, Pittsburgh, USA, 2002, pp. 161–166. [Online]. Available: citeseer.csail.mit.edu/548531.html

[37] C. J. Cohen, L. Conway, and D. Koditschek, "Dynamical system representation, generation, and recognition of basic oscillatory motion gestures." in *International Conference on Automatic Face- and Gesture-Recognition*, 1996.

[38] R. E. Kahn, M. J. Swain, P. N. Prokopowicz, and R. J. Firby, "Gesture recognition using the perseus architecture," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1996.

[39] D. C. Knill and A. Pouget, "The bayesian brain: the role of uncertainty in neural coding and computation," *TRENDS in Neurosciences*, vol. 27, pp. 712–719, 2004.

[40] G. E. Loeb, "Learning from the spinal cord," *Journal of Physiology*, vol. 533.1, pp. 111–117, 2001.

[41] J. Rett and J. Dias, "Visual based human motion analysis: Mapping gestures using a puppet model," in *Proceedings of EPIA 05, Lecture Notes in AI, Springer Verlag, Berlin*, 2005.

[42] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, p. 511. [Online]. Available: http://www.ai.mit.edu/people/viola/

[43] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[44] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, no. Q2, p. 15, 1998. [Online]. Available: citeseer.ist.psu.edu/bradski98computer.html