FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D3.3.1
Set-up of pan/tilt head and stereo-camera for tracking humans and detecting their desire to interact

Due date of deliverable: 14/1/2004
Actual submission date: 20/1/2004

Start date of project: January 1st, 2004 **Duration:** 48 months

Organisation name of lead contractor for this deliverable:
Vrije Universiteit Brussel

Revision: final
Dissemination Level: PU

Executive Summary

This report describes the software and hardware of an experimental setup to study a component of a cognitive robot for detecting and tracking humans and for reading their willingness to interact. The aim of these setup and the related experiments is to build a system which enables a robot to reliably detect and track humans using multiple visual cues, such as chromatic data, depth, motion and facial features. On top of the tracking mechanism, a number of heuristics report on the willingness of the user to interact with the robot. The system has been designed to be robust to different lighting conditions, it can handle different poses of the user, it is robust to the distance that the user maintains to the robot and can handle multiple users simultaneously. This last feature is implemented as an attention mechanism, where the robot keeps its attention with one interested user without being distracted by others or other distractions. The setup consists of a stereo colour camera mounted on a pan/tilt head and runs off a laptop computer, insuring is portability.

Role of this experimental setup in Cogniron

This research fits within WP3.3 Understanding willingness to interact, which fits within WP3 Social behaviour and embodied interaction. If a cognitive robot is to interact with humans, it should first of all be able to detect the presence of humans. However, this alone is not enough. It should also be aware of when humans are willing to interact. Humans are excellent at predicting whether someone wishes to engage in an interaction, such as friendly chat, and base their judgement on several cues such as position, pose, and gaze. The research in WP3.3 wishes to implement this behaviour on a cognitive robot.
This work is integrated in a set-up for learning concepts through linguistic interactions at the VUB (Vrije Universiteit Brussels) and is also intended to be used by UH (University of Hertfordshire) in their studies on interaction style and personal spaces (see WP 3.1).

Relation to the Key Experiments

The research in WP3.3 fits within Key Experiment 2 The curious robot, but will also be of use in Key Experiment 1 Home tour.

1 Introduction

1.1 Social cognition

When designing a cognitive robot, one has to make sure not to overlook social mechanisms that we humans take for granted and that are therefore often ignored during the design of a social robot companion. Humans continuously and effortlessly assess the stance of others, and are masters at reading social cues. This is related to the Theory of Mind (ToM), well known in psychology and cognitive science . A ToM mechanism observes humans and attributes beliefs and desires: the mechanism has a grasp of what people do and feel [Baron-Cohen, 1994]. The willingness to interact is an important aspect of this. ToM concepts already appear very early in life, at an age where abstract reasoning and knowledge is hardly developping yet. This, together with neurological evidence such as mirror neurons [Rizzolatti et al., 1996], suggests that a ToM mechanism is crucial for functioning in a complex social environment, and therefore a mechanism which perform similarly will be central in any cognitive robot.

Work in human-computer interaction mainly focuses on how humans, through different modalities, can control and interface with computers and machines [for an example see Dix et al., 2003]. Computers are in this context still seen as tools, not much different from a dishwasher or a burglar alarm, and not as technological companions. The objective of the Cogniron project is to study components of a *cognitive companion*, meaning that the project develops methods and technologies to build a robot which behaves as a companion, a companion which possibly acts as an aid for a number of chores, but which should never be considered a mere tool. Tools require an exact and never faltering interface, but a cognitive robot companion should be addressed as just another human being, using verbal and non-verbal communication with all the related imprecission and ambiguity.

The technological setup and the algorithms presented here are not unique. Numerous implementations exist for detecting and tracking faces and human figures, for recognising and interpreting gestures, for recognising and reading faces, and so forth. Even though the setup we use is state-of-the-art, the approaches taken do not exemplify a break through in field of vision based interpretation of human activity. What is refreshing is the view we take on human-robot interaction. We aim not at building a man-machine interface to *control* robots; instead we want the robot to *interact* with its users in a natural way: humans also do not interact which each other with the aim to control the other, and even if they do, their interaction is not to be compared with any man-machine interaction mechanism known to the HCI community. The starting point of our design therefore is radically different than most man-machine interfaces.

1.2 Linguistic interaction

The set-up as described in the report was in part designed to be used for learning concepts of objects and spaces through linguistic interactions. Concept formation, and the related field of the evolution of language and its cognitive components [for a sample of relevant papers see Cangelosi and Parisi, 2001], has recently received much attention from the cognitive science and artificial intelligence community. At the Vrije Universiteit Brussel we study how categories and concepts can be learned through *linguistic interactions*. The construction of categories, concepts and ontologies has often relied on either hand-crafting or on unsupervised inference from a set of training data. For several reasons it is now clear that these approaches are severely limited, and will never suffice to obtain mental representations that are coordinated with human mental representations. We believe that categories, concepts and ontologies should be acquired through a continuous dialogue between the robot and its users

[Steels and Belpaeme, 2005]. Many approaches have already been detailed which could facilitate this process [Steels and Kaplan, 1999, 2002, Vogt, 2000, Belpaeme, 2004, 2005]. These approaches rely on a simple word exchange between two agents describing a context of objects to each other, during which the agents shape their representations based on the feedback they receive from using particular words in particular contexts. These interactions have, even though they have always been grounded up to some extent [Harnad, 1990], not convincingly been shown to work for complex interactions with humans. The research presented in this report describes the first steps towards constructing an experimental platform for linguistic interaction between a robotic agent and humans.
Dialogue draws heavily on shared attention and social cues between the dialogue partners (even though modern media of dialogue, such as telephoning and chatting, tend to obfuscate this point). The platform described here implements one of the social capabilities necessary for dialogue between robots and humans: the ability to read the willingness to interact.

1.3 Related work in detection and recognition of humans

The research reported here focuses on visual cues for reading dispositions and intentions. Although other cues might also be used to read social behaviour, visual cues are most prominent as they are easier to read, have low noise and carry a long distance, as opposed to, for example, verbal cues. Instead of wanting to replicate a faithful neurophysiological model of social cognition, we take a pragmatic approach in designing a social robot. Starting from available hardware and requirements posed by a mobile robotic system, we have built an experimental setup which uses a number of readily available cues to make an estimate of the users desire to interact. In this, position and pose of the user play an important role, but the most important role is reserved for the face and its orientation. Psychologists (such as [Young, 1998]) have already described the importance of facial cues in reading social behaviour, and it is therefore not surprising that also in the functioning of our system facial information is crucial. Although the system does not read gaze direction or emotional expressions, it is able to reliable predict whether a user is interested in the robot and thus desires interaction.
There exists a vast amount of literature concerning the detection and recognition of humans in images and video. Therefore we will only glance at some related work that resembles our setup most. In [Darrell et al., 1998] a person tracking system is proposed which makes use of stereo vision, color information and a pattern detector for faces. [Morency et al., 2002] describes a system for head tracking and head-pose estimation which makes use of a technique called stereo-motion head tracking. Recently, in [Seemann et al., 2004], a similar system is presented which, apart from stereo vision and face detection, makes also use of color, for tracking human heads.
The system we presented in this report does not necessarily improve upon these previous implementations, but was built with a specific function in mind: the detection of human's interest in the robot and its willingness to interact. It resembles each of the described systems in one or more of the following ways: it works in real-time, in an uncontrolled environment and uses stereo vision, motion, color and face pattern detection as input modalities. On the other hand, at this time, we lack the need for very precise head-pose estimation (as is e.g. the case in a hands-free control equipment), which was therefore not implemented. Another difference is that, as far as we know, the mentioned person/head trackers make use of a fixed camera, while our stereo camera is mounted on a pan-tilt head. This significantly increases the 3D volume in which the tracker can operate, but also makes the tracker slightly more complicated, as position and motion information also depend on the position and speed of the pan-tilt head.

2 Description of the System

Physically, the system consists of three main components. First we have a portable computer, with 512 Mb ram and an Intel Pentium M processor at 1500 MHz. Second, we use the commercialy available SVS (small vision system) from SRI International. This is a stereo vision system which, in our case, consists of a MEGA-DCS stereo head, with two color cameras, and accompanying software for depth calculation. We use lenses with 12 mm focus. The stereo head uses the IEEE 1394 port. Finally, this stereo head is mounted on a TrackerPod, which is a pan-tilt unit from TrackerCam, which is controlled via a USB interface.
The software system, which runs on the laptop, currently takes as input the images coming from the stereo head, and produces output by controlling the pan-tilt head, and by giving visual feedback about its internal state and running processes on the computer screen. Of course, both these inputs and outputs could be extended when the system is integrated with other systems (e.g. a mobile robot, a speech recognition/production system, etcetera).
The following section gives an overview of the different input modalities that are used. Subsequently, the processing of these inputs is discussed in detail.

2.1 Input channels

In the following we will use the following notation, at time t:

- $I(t)$ is the grayscale image from the left camera,
- $C(t)$ is the BGR-color image from the left camera,
- $D(t)$ is the disparity image, resulting from the stereo vision process, and gives the horizontal distance in 1/16 pixel unit between corresponding pixels in the left and right image

All images have size 320×240.

2.1.1 Motion

If the pan-tilt head is not moving, motion in the image can give an indication in which regions interesting objects, especially humans, can be found. Therefore, at each timestep t, a binary image, the motion mask M_m, is computed as follows:

$$M_m(t) = |I(t) - I(t-1)| > \delta_m,$$

in which the absolute value $|.|$ and comparison $>$ operator act upon every image pixel. In our system δ_m was set to 20.

2.1.2 Color

Another source of information is the chromatic content of the images. Although color constancy is an unsolved problem and skin color can vary significantly between individuals and in changing light condition, we nevertheless opted for using a rather weak skin color detector, in the sense that most of the time a lot more than only skin color is detected.

First the RGB color values from the left camera are converted to the HSV color space:

$$\begin{aligned} V &= \max(R,G,B) \\ S &= \begin{cases} 255\frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \\ H &= \begin{cases} 60(G-B)/S \bmod 360 & \text{if } V = R \\ 180 + 60(B-R)/S \bmod 360 & \text{if } V = G \\ 240 + 60(R-G)/S \bmod 360 & \text{if } V = B \end{cases} \end{aligned}$$

Then, using this color space, the color mask C_m is determined as follows:

$$\begin{aligned} C_m = \; & H(C) \in \text{Circ}(h_l, h_u) \wedge \\ & S(C) \in [s_l, s_u] \wedge \\ & V(C) \in [v_l, v_u]. \end{aligned}$$

In which

$$\text{Circ}(a,b) = \begin{cases} [a,b[& \text{if } a \leq b \\ [b,360[\cup [0,a[& \text{if } b < a \end{cases}$$

In our system the following values were used: $h_l = 8$, $h_u = 48$, $s_l = 65$, $s_u = 205$, $v_l = 89$, $v_u = 256$.

2.1.3 Stereo vision

As was mentioned before, the software that calculates a depth map by searching for corresponding pixels in the right and left image, is part of the SVS. We briefly discuss the most important parameters for stereo calculation and their settings.
One parameter is the side of the square window which is matched between the two images. In our system this window had a size of 11×11. Another parameter is the number of disparities that is searched for, which was fixed at 48 (pixels). One can also define a horizontal offset (x offset) which shifts the search range. This offset, together with the number of disparities, define the horopter: the 3D volume in which the stereo algorithm functions. Unlike the search range which is fixed, we controlled the x offset continuously in order to keep the object at hand (probably a human head) well within the bounds of the horopter.
Once a depth map is built, the SVS software allows to determine the 3D positions, in the camera coordinate frame, of individual points. Inversely, one can also calculate the projection of a point, given its 3D coordinates.
The coordinate frame attached to the left camera has its origin on the focus point of the lens, the z-axis collides with the optical axis of the lens and has positive values in front of the camera. The x-axis lies horizontally and has positive values to the left, when facing the camera. The y-axis lies vertically and its values increase going down. The unit of length is mm.

2.1.4 Face Detection

The system makes extensive use of OpenCV [opencv], an open source computer vision library. This library contains an object detection algorithm, 'a boosted cascade of simple features' [Viola and Jones,

2001, Lienhart and Maydt, 2002], and the necessary data for detecting frontal faces. Using this data, the algorithm decides whether a certain part of the image corresponds to a face. For performance reasons it is not possible to scan the entire image for faces of different sizes every image frame. Though, once some more information is known about the position of a face, this technique can be used very effectively, as will be discussed in the next section.

2.2 Processing

In this part will be discussed how the different input modalitities: color, motion, depth and face detection, are integrated. Central in the processing algorithm is the concept of a *head state*, in which all the information about a possible human head is stored. The state of the whole system is uniquely determined by a set of such head states.
The system runs three different processes simultaneously. The exploration process scans the 3D space covered by the vision system in order to detect new possible human heads and create new head states. The updating process adapts the known set of head states according to new information and possibly removes head states that are no longer reliable. Finally, the controlling process categorizes the set of head states according to reliability and estimated interest in the robot of the presumed human and controls the pan-tilt head to track the most interesting subject. During the description of the different processes, auxiliary functions will be defined when necessary.

2.2.1 The head state

In the following, we define the average with forgetting rate α of a time-sequence X as $\widehat{X}$, with $\widehat{X}_0 = X_0$ and $\widehat{X}_i = (1-\alpha)\widehat{X}_{i-1} + \alpha X_i$.
A head state consists of the following components:

- $T = \{t_x, t_y, t_z\}$: the 3D position of the center of the head (in mm)
- $V = \{v_x, v_y, v_z\}$: the 3D velocity of the head (in mm s^{-1})
- $\widehat{V}$: the average velocity of the head, using a forgetting rate α_v (in mm s^{-1})
- Δ_{motion}: time since the last detected motion (in s)
- Δ_{face}: time since the last detected face (in s)
- Δ_{faceTry}: time since the last try to detect a face (in s)
- $\widehat{m}$: average motion detection rate with forgetting rate α_m, where

$$m = \begin{cases} 1 & \text{if there is motion in the image region of the head} \\ 0 & \text{otherwise} \end{cases}$$

- $\widehat{f}$: average face detection rate with forgetting rate α_f, where f is defined analoguously as m
- age: time since creation of the head state (in s)
- $\Delta_{\text{inv(isible)}}$: time since last successful update of the head state (in s)
- ID: a number which is unique among all head states.

In the system the parameters were set as follows: $\alpha_v = 0.05$, $\alpha_m = 0.03$, $\alpha_f = 0.3$. As will be explained in the next section, face detection is not done at every timestep, but only at regularly intervals. This is why α_f is chosen significantly bigger.

2.2.2 The Updating Process

This process updates every head state, part of the current state, according to newly available information coming from the previously described channels. It also decides whether a head state should be kept.
Let Δ be time since the last update ($\approx 1/$framerate). A rectangle (in the image plane) is specified by its upper left ($a = [a_x, a_y]$) and lower right ($b = [b_x, b_y]$) corner and denoted as $[a, b]$.
At first the following auxiliary variables are calculated:

$$\begin{aligned}
T' &= T(t) + \Delta V(t) \\
b &= \begin{cases} 2b_0 & \text{if panTiltMoving} \\ b_0 & \text{otherwise} \end{cases} \\
\text{project}(P) &= \text{gives the (left) image coordinates of 3D point } P \\
\text{faceRect}(P) &= [\text{project}(P + [-W/2, -H/2, 0]), \text{project}(P + [W/2, H/2, 0])] \\
\text{searchRect}(P) &= [\text{project}(P + [-W/2 - b, -H/2 - b, 0]), \text{project}(P + [W/2 + b, H/2 + b, 0])] \\
B_{\text{rect}} &= \text{faceRect}(T') \\
S_{\text{rect}} &= \text{searchRect}(B_{\text{rect}}).
\end{aligned}$$

in which T' is a first estimation of the new head position. B_{rect} is a first approximation of the new rectangle the head covers in the image. The head width and height W and H were set to 140 and 168 respectively (in mm). S_{rect} is the rectangle in which the head is assumed to lie and in which restricts the search area for subsequent processes. Its size depends on b, which specifies an extra border (in 3D space) around the head, and which is chosen larger when the pan-tilt head is moving. b_0 was set to 66 (in mm). If B_{rect} does not lie within the visible window, then $\Delta_{\text{inv}}(t+1) = \Delta_{\text{inv}} + \Delta$ and the update process aborts. If moreover $\Delta_{\text{inv}}(t+1) > \delta_{\text{inv}}$, the head state under consideration is removed. The parameter δ_{inv} was set to 1.0 (in s).
Face detection is applied regularly, more precisely if $\Delta_{\text{faceTry}} \geq \delta_{\text{faceTry}}$. In that case we calculate

$$\begin{aligned}
\text{detectFace}(I, r) &= \text{if successful, the rectangle surrounding the detected face in rectangular area } r \\
F_{\text{rect}} &= \text{detectFace}(I, S_{\text{rect}}).
\end{aligned}$$

Δ_{face}, Δ_{faceTry} and $\widehat{f}$ are updated in a straightforward way, depending on whether face detection was tried and if so, whether a face was found.
In the more frequent case that no face detection is performed, the following calculations take place:

$$\begin{aligned}
\text{filter}(D, a, b) &= \text{from the depth image } D\text{, the binary image indicating the points} \\
&\quad \text{with depth between } a \text{ and } b \text{ (in mm)} \\
K \wedge L &= \text{the binary image which is the logical and of binary images } K \text{ and } L \\
J &= C_m \wedge \text{filter}(D, T'_z - d, T'_z + d))
\end{aligned}$$

such that J indicates the points with skin color and more or less correct depth. The depth range d was set to 300 (in mm).

Subsequently, F_{rect} is now determined as follows:

$$\begin{aligned} \text{maxRect}(A, r, w, h) &= \text{the rectangle with size } w \times h, \\ &\quad \text{that covers most pixels of binary image } A \\ F_{\text{rect}} &= \text{maxRect}(J, S_{\text{rect}}, \text{width}(B_{\text{rect}}), \text{height}(B_{\text{rect}})) \\ &. \end{aligned}$$

Once this new estimated face rectangle (F_{rect}) is known, a new estimated 3D position can be derived, as well as a new depth mask:

$$\begin{aligned} \text{calc3DPos}(D, r, A(optional)) &= \text{gives the average 3D position of the points} \\ &\quad \text{within rectangle } r\text{, optionally filtered by binary image A} \\ T'' &= \text{calc3DPos}(D, F_{\text{rect}}) \\ K &= \text{filter}(D, T''_z + d_{\text{further}}, T''_z - d_{\text{closer}}) \\ &. \end{aligned}$$

with d_{further} and d_{closer} respectively equal to 70 and 200 (in mm).
Next, the new head position is determined as follows, together with the velocity:

$$\begin{aligned} T(t+1) &= \text{calc3DPos}(D, \text{searchRect}(F_{\text{rect}}), K) \\ .V(t+1) &= \frac{T(t+1) - T(t)}{\Delta} \end{aligned}$$

The quantities $\widehat{V}$ as well as age are updated straightforwardly.
It is possible that calc3DPos failed during one of the previous steps, due to an insufficient number of reliable depth pixels. In that case the same action is taken as when B_{rect} did not lie within the visible window.
Finally, the motion-related values Δ_{motion} and $\widehat{m}$, are updated, but only if the pan-tilt head is not moving:

$$\begin{aligned} \text{count}(A, r) &= \text{the number of on-pixels within rectangle } r \text{ in binary image } A \\ \text{motion} &\Leftrightarrow \text{count}(M_m, \text{faceRect}(T(t+1)) > \delta_m \end{aligned}$$

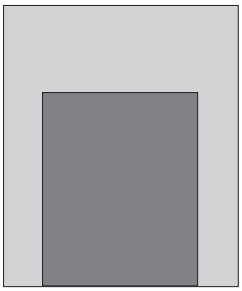
with $\delta_m = 20$ (number of pixels).

2.2.3 The Exploration Process

The previous section described how existing head states were updated, but nothing was said about how these head states were created in the first place. This is the task of the exploration process, which scans the visual space for possible human heads and creates new head states for them. When scanning at a certain depth r (in mm), the process first performs the following calculation:

$$J = C_m \wedge \text{filter}(D, r - \delta_r, r + \delta_r)$$

such that J is a mask indicating the pixels with correct color and depth, with $\delta_r = 250$ (in mm). Next, in J is searched for regions mathing the following binary template:



which is matched best when all the pixels in the dark rectangle are 1 and all the others 0. A region matches when the fraction of mathing pixels exceeds a threshold δ_f, which was set to 0.15. The size of the template depends on the depth r, such that the dark rectangle has the expected size of a human head at depth r. For each matching region the position of the dark rectangle is stored. Next, the obtained rectangles are filtered to remove overlapping rectangles. In case of an overlap, the rectangle that resulted from the best match is kept. Furthermore, the 3D position of the rectangles is determined using calc3DPos and finally, if the depth could be reliably estimated, a head state is created containing the appropriate values.
The previous process is repeated cyclicly at different depths, in steps of 250 mm, ranging from 400 mm to 10000 mm from the camera.

2.2.4 The Controlling Process

The controlling process has to decide which head states are likely to correspond to a human head, and which, if any, of the head states indicate that the person is interested in the robot.
Based on that information, this process also controls the pan-tilt head and performs the necessary coordinate transformations on the T, V and $\widehat{V}$ component of each head state, in order to compensate for the change in pan-tilt position. V and $\widehat{V}$ are changed only taking into account the orientation of the pan-tilt head, not its velocity, as this is difficult to estimate reliably.
To accomplish the previous tasks, the control process first sorts the different head states according to the following procedurally defined order relation (a dot is used to access components of a head state):

$$
\begin{aligned}
\text{greater}(a,b) \;=\; & \texttt{if } (a.\Delta_{\text{face}} < \gamma_{\text{face}} \wedge b.\Delta_{\text{face}} < \gamma_{\text{face}}) \\
& \quad \texttt{if } (|a.\Delta_{\text{face}} - b.\Delta_{\text{face}})| > \beta_{\text{face}}) \\
& \quad\quad \texttt{return } a.\Delta_{\text{face}} < b.\Delta_{\text{face}} \\
& \texttt{if } (|a.T_z - b.T_z| > \beta_z) \\
& \quad \texttt{return } a.T_z \leq b.T_z \\
& \texttt{if } (|\,\|a.V\| - \|b.V\|\,| > \beta_V) \\
& \quad \texttt{return } \|a.V\| < \|b.V\| \\
& \texttt{if } (a.\Delta_{\text{motion}} < \gamma_{\text{motion}} \wedge b.\Delta_{\text{motion}} < \gamma_{\text{motion}}) \\
& \quad \texttt{if } (|a.\Delta_{\text{motion}} - b.\Delta_{\text{motion}}| > \beta_{\text{motion}}) \\
& \quad\quad \texttt{return } a.\Delta_{\text{motion}} < b.\Delta_{\text{motion}} \\
& \texttt{return } a.\text{age} > b.\text{age}
\end{aligned}
$$

with $\gamma_{\text{face}} = 20$(s), $\gamma_{\text{motion}} = 7$(s), $\beta_{\text{face}} = 5$ (s) and $\beta_{\text{motion}} = 3$(s).
Next, this ordered list is filtered to remove head states that are closer than $\delta_s = 50$(mm) to each other, giving preference to head states appearing earlier in the list.

Furthermore, given the current position of the pan-tilt head, the positions T in camera frame coordinates are tranformed into world coordinates (fixed with respect to the pan-tilt base). Based on this world coordinates, head states which occupy an unlikely position (too high or too low), are also removed.
Finally, only a certain number of head states is kept, in our case five.
The control process can treat one head state special if it is believed it corresponds to a person interested in the robot. The criterion for a head state to get that special status is the following:

$$\begin{aligned} \sqrt{\widehat{V}_x^2 + \widehat{V}_y^2} &< \eta_{\text{lat}} \\ \widehat{V}_z &< \eta_{\text{front}} \\ \widehat{f} &< \theta_{\text{face}}, \end{aligned}$$

with $\eta_{\text{lat}} = 50$ (mm s^{-1}), $\eta_{\text{front}} = 20$ (mm s^{-1}) and $\theta_{\text{face}} = 0.5$. V is expressed in the camera coordinate frame. The criterion for not losing that special status is the same, except that some hysteresis is built in by relaxing the inequalities with a certain factor, which in our case was set to 2. If a head state is treated special it is also tracked.
Even if no head state is considered special, there is still another criterion by which the control process can select a head state to track:

$$\begin{aligned} \widehat{m} &> \mu_{\text{motion}} \\ \widehat{f} &> \mu_{\text{face}}, \end{aligned}$$

with $\mu_{\text{motion}} = 0.5$ and $\mu_{\text{face}} = 0.1$.
If a head state to track is selected, in one way or another, the image projection T project(T) is calculated and if the x or y coordinate deviates more than 10%, relative to the image size, from the image center, the pan and tilt positions are changed accordingly.

2.3 Results and conclusion

For a demonstration of the experimental setup the reader is referred to the Cogniron D3.3.1 demonstration video. The system has been qualitatively tested at the Artificial Intelligence Lab of the Vrije Universiteit Brussel and in a home environment under different circumstances of lighting, position and background. The system has shown its robustness during the following situations.

- A single individual as well as multiple individuals are tracked, the attention mechanism takes care of focussing the attention of the system on only one individual at a time.
- People walking by, who not interested in the robot, are not picked up by the tracker. Moving objects are not falsely assumed to be humans.
- Occlusions of the face do not disturb the working of the system. The video for example shows a test subject drinking from a glass while the system still tracks his face, although during drinking the system does not consider the subject to be interested in the robot.
- Small and large skin coloured regions which are not human faces do not confuse the system. The video shows how the floor of one of our test environments, which is classified as being skin coloured, is not picked up by the tracker.

- Face-like images which are not true human faces are not picked up by the tracker. The video shows how a cartoon version of a face on a sheet of paper is correctly not identified as a human face. Other experiments, not shown in the video, show that the system does pick up faces on a realistic painting.

The simple heuristics for deciding if the user is interested are rather rules-of-thumb than based on psychological data, but nevertheless seem to perform satisfactory. The heuristics could potentially benefit from psychological input; information which might be provided by the University of Hertfordshire.

3 Future Work

According to Baron-Cohen [1994] a mindreading system consists of four parts; (1) an intentionality detector, (2) an eye-direction detector, (3) a shared-attention mechanism and (4) a Theory of Mind mechanism. Each can be considered as a cognitive module, which sets an agenda for further progress in building a cognitive robot.
The research described in this report has gone some way in constructing the first part of Baron-Cohens route towards social cognition. A logical progression would be to extend this research and implement an eye-direction detector. Moreover, recently cognitive scientist have stressed the importance of eye gaze and eye movement as a social cue [Langton et al., 2000]. Indeed, as humans are one of the only species having clearly visible eye white, which facilitates the reading of gaze direction, this seems to be a too important cue not to have on a cognitive robot.
Equally important, but relying on face and gaze direction and therefore only implementable after we have an eye-detection module, is the shared-attention mechanism. Shared attention aids the robot in understanding communication and instruction, and will be crucial for a socially situated system.

4 References

4.1 Applicable documents

Not applicable.

4.2 Reference documents

S. Baron-Cohen. How to build a baby that can read minds? *Cahiers de Psychologie Cognitive*, 13: 513–552, 1994.

Tony Belpaeme. Does structure in the environment influence our conceptualization? Poster presentation at the Fifth International Conference on the Evolution of Language (EVOLANG 2004), Leipzig, Germany, 2004.

Tony Belpaeme. Robots should not be robinson crusoes. Paper given at the AI in the Wild Symposium, Groningen, The Netherlands, 2005.

Angelo Cangelosi and Domenico Parisi, editors. *Simulating the Evolution of Language*. Springer Verlag, London, 2001.

T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 601. IEEE Computer Society, 1998. ISBN 0-8186-8497-6.

Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human Computer Interaction*. Prentice Hall, 2003.

Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

Stephen R.H. Langton, Roger J. Watt, and Vicki Bruce. Do the eyes have it? Cues to the direction of social attention. *Trends in Cognitive Science*, 4(2):50–59, 2000.

Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing*, pages 900–903, Rochester, USA, September 2002. IEEE.

Louis-Philippe Morency, Ali Rahimi, Neal Checka, and Trevor Darrell. Fast stereo-based head tracking for interactive environments. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 390. IEEE Computer Society, 2002. ISBN 0-7695-1602-5.

opencv. Intel. Open Source Computer Vision Library (OpenCV), http://www.intel.com/research/mrl/research/opencv/.

G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.

Edgar Seemann, Kai Nickel, and Rainer Stiefelhagen. Head pose estimation using stereo vision for human-robot interaction. In *FGR*, pages 626–631, 2004.

Luc Steels and Tony Belpaeme. Coordinating perceptually grounded categories through language. A case study for colour. *Behavioral and Brain Sciences*, 2005. Accepted as target article.

Luc Steels and Frédéric Kaplan. Bootstrapping grounded word semantics. In T. Briscoe, editor, *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press, Cambridge, 1999.

Luc Steels and Frédéric Kaplan. AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32, 2002.

Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518, 2001.

Paul Vogt. *Lexicon grounding on mobile robots*. PhD thesis, Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium, 2000.

A.W. Young. *Face and Mind*. Oxford University Press, Oxford, UK, 1998.

Annexes

Not applicable.