# AIBO ROBOT AS A SOCCER AND RESCUE GAME PLAYER

**Dragos Datcu, L.J.M. Rothkrantz**
**Faculty of Electrical Engineering, Mathematics and Computer Science**
**Delft University of Technology**
**Mekelweg 4, 2628 CD Delft, The Netherlands**
**E-mail: { D. Datcu, L.J.M. Rothkrantz }@ewi.tudelft.nl**

## KEYWORDS

Situational AI, Domain knowledge specification and character instruction, Pathfinding.

## ABSTRACT

The researches in the field of robotics technologies are set to improve the way intelligent processes accomplish their tasks and interact with the environmental world.
An efficient mode for researching and testing the capabilities of the algorithms is putting the robots to play in different coordination games. The current project aims to set up an environment for Sony AIBO robots to play in a soccer game and a rescue game. In both games AIBO is assumed to detect and localize objects such as balls, shouting team players, landmarks and crying victims. The AIBO robot is able to sense the world with different modalities, i.e. sight, hearing and touch. The ultimate challenge of the paper is to fuse data from different communication channels.
An additional approach is to program the robots for playing the Rescue game. The results relate to valuable information on common safety and emergency management issues.

## INTRODUCTION

In playing games, the robotic systems automatically extract meaning from multimodal, raw input data, and conversely produce perceivable information from symbolic abstract level.
By setting the robots for playing in games with high difficulty levels, all the involved algorithms and the specific behavior at different interaction levels can be evaluated. During a common game session, the robot's hardware components can be programmed to execute intricate actions. Commonly, the actions are backed by a previous reasoning step that takes into account data from different channels. Information fusion from sensors plays an important role in combining the actual discovery and the previous knowledge for taking any action. In the current research we have involved a multi-agent system approach for cooperation and learning for legged robots. A special case concerns real-time control algorithms for AIBO football playing robots. The final result is a modular architecture that allows flexible implementation and future extension.
The efforts have been mainly focused on sound, vision, movement and gait generation. Gait generation stands for a useful capability of a great importance in many practical aspects.
Multimodal systems represent and manipulate information from different sensor channels at multiple levels of abstraction.
The human sensory system perceives processes and fuses multimodal information provided by multiple modalities and enables us to interpret the resulting information with an apparent ease within the current context (time, environment, history). It handles the problems of ambiguity, redundancy and synchronicity in a seamless s manner. Nevertheless, it is prone to the problem of information overload and human observers are not available at any place and at any time.
In the proposed project information will be derived from sensors measuring different modalities. The resulting intelligent system must have an awareness of the spatial properties of the environment, changes of those properties over time and it must have an awareness of the behavioral aspects. The general question is how the data from those sensors can be fused in order to obtain a consistent and robust model of the world.
The fusion can be realized on different levels, at the close to signal data level or at the highest semantic level. Fusion at the basic level has a general character and can be used in many applications. Especially cueing that is the expected sound triggering of visual attention is suited for low-level fusion. Also the fusion of information from spatially distributed sensors is a good candidate for success at the lowest level.
Information fusion at the highest level implies context awareness and modeling the patterns of behaviour. To summarize, in the current project, information will be derived from sensors measuring different modalities sound and vision. Our intelligent system must have an awareness of the spatial properties of the environment (locate objects and people, build maps), changes of

those properties over time (tracking objects and people) and it must have an awareness of the behavioral aspects (recognize intentions and emotions). The general question is how the data from those sensors can be fused in order to obtain a consistent and robust model of the world.

## RELATED RESEARCH

[J.C.Martin et al. 1995] describes a basic language for cooperation of modalities in a multimodal application. The event queue mechanisms and the communication between tasks and execution modules are also analyzed.
[M.T.Vo, A.Waibel 1997] presents a toolkit for multimodal application development. It includes graphic tools that enhance multimodal component and integration modules with context free grammar support.
Krahnstoever [N.Krahnstoever 2001] describes a multimodal framework targeted specifically at fusing speech and gesture with output being done on large screen displays. Several applications are described that have been implemented using this framework. The fusion process is not described in great detail, but appears to be optimized for and limited to integration of speech and gesture, using inputs from cameras that track a user's head and hands.
The W3C has set up a multimodal framework specifically for the web [J.A.Larson, T.V.Raman 2002].
This does not appear to be something that has actually been implemented by the W3C. Rather, it proposes a set of properties and standards specifically the Extensible Multimodal Annotation Markup Language (EMMA) that a multimodal architecture should adhere to.
The field of multi-sensory data fusion has witnessed a number of advances in the past couple of years, spurred on by advances in signal processing, computational architectures, and hardware. Nevertheless, these advances pertain mainly to the development of different techniques (e.g., Kalman fusion, ANN-based fusion, HMM-based fusion) for multimodal data fusion at a single abstraction level. To date, most multi-sensory data fusion approaches have largely avoided dealing with questions that involve context -awareness and internationality and, in turn, require the realization of multimodal data fusion at different abstraction levels.
Existing approaches to multi-modal data fusion are usually limited to multi-sensory data fusion at a single level. Multi-modal fusion in literature has a strong focus on improving speech recognition and person identification. In the past couple of years our research group was already involved in multi-modal fusion projects [M.Pantic, L.J.M.Rothkrantz 2000, 2003].
At the Delft University of Technology has been a project running on the development of a multimodal framework for multimodal data fusion [D.Datcu, L.J.M.Rothkrantz 2004]. On the project, the data were supposed to be gathered through distinct media channels.

## AIBO robot’s hardware platform

The processing power of the AIBO (ERS-7 version) robot is supported by a 576 MHz 64bit RISC CPU. The built-in Wireless LAN module (IEEE 802.11b) offers communication capabilities within the network with computers or other robots. All the data processed by the robot are stored on the existent internal memory of 64 MB or on the attached flash memory. The robot is equipped with a color vision camera that is a CMOS Image Sensor capable of capturing images of 350.000 pixels.
The sound is handled through incorporated miniature microphones and speaker. The infrared distance sensors mounted next to the vision camera and on the main body provide the robot with data related to the obstacles on the path. In combination with the vision camera, all the present sensors (Table 1) are set to offer the main processing unit of the robot detailed information concerning the perception of the surroundings on different channels.

**Table 1. Hardware details of the AIBO ERS-7**

| | |
|---|---|
| **CPU:** | 64bit RISC Processor, MIPS R7000 |
| **CPU Clock:** | 576MHz |
| **Movable Parts:** | Mouth - 1 degree of freedom<br>Head - 3 degrees of freedom<br>Leg - 3 degrees of freedom x 4 legs<br>Ear - 1 degree of freedom x 2<br>Tail - 2 degrees of freedom |
| **Built-in Sensors:** | Temperature Sensor<br>Infrared Distance Sensor (head, body)<br>Acceleration Sensor<br>Electric Static Sensor (head, back)<br>Pressure Sensor (chin, paws)<br>Vibration Sensor |
| **Operating Time:** | Approx. 1.5 Hours (Standard operation) |
| **LED:** | |
| - Illume Face: | 28 LED (white 16, red 4, blue 4, green 4) |
| - Ear : | 2 (left and right) |
| - Head sensor : | 2 (white and amber) |
| - Head (WLAN) : | 1(blue) |
| - Back sensor : | 16 (white 8, red 3, blue 3, orange 2) |

In order to program the robot for playing a specific game, first a good knowledge of the corresponding tasks has to be obtained.
All the actions that have to be taken by the robot can be classified through a number of processing layers. At the base layer there are the tasks consisting of low level processing routines focused on the input signals received from the robot’s sensors.

## FUNCTIONAL ARCHITECTURE

In the case of playing in a football game, the AIBO robot has to perform specific tasks according to the current running context. All the actions are taken according to a finite state machine model. The program contains sets of premises for entering one of the states and rules for specifying the actions. Some of the states may include specific motion actions of the robot’s components. The other states are strictly related to reasoning procedures for the given scene.

While playing in a soccer game, every robot has to send information related to its own perception and processing to the other playing robots in its team. The information set to be sent contains also the location of the robot and that of the ball. In case one robot does not have a clear view of the ball so as to establish its location in the game field, it uses the approximation given by the other robots.

## PROGRAMMING AIBO ROBOT

The AIBO (ERS-7) robot is highly programmable. In order to write applications that directly interact with robot’s hardware resources, some requirements have to be fulfilled. The programs can be developed using special tools and compilers. There are also routine libraries for making the process possible. AIBO SDE is an AIBO Software Development Environment that can make software that either executes on AIBO, or executes on a PC and controls AIBO by using a wireless LAN.

The AIBO SDE package contains three SDKs and one motion editor. The first SDK is OPEN-R and is a cross development environment based on gcc (C++) that can be used for making software that works on AIBO. The SDK also provides a powerful scripting language for efficiently implementing the algorithms.

It is also possible to write applications that have access to the "OPEN-R system layer" and so to have direct control to the basic functions of the robot. Because of its capabilities, the OPEN-R SDK represents a suitable tool for developing application in the field of robotics research.

The second SDK is R-CODE and contains an environment that offers the possibility to execute programs written in R-CODE, a scripting language, on AIBO (ERS-7).

The third SDK is the AIBO Remote Framework and runs in Windows PC environments. The application stands for a development tool based on Visual C++ that can be used for making software that runs on a Windows PC. The AIBO Remote Framework contains a server program, libraries and header files for the client application.

It is possible to remotely give commands to AIBO (ERS-7) robot via wireless LAN through the developed applications. AIBO Motion Editor is a tool for creating motion editor schemes for the robot.

## SOUND SOURCE LOCALIZATION

While playing soccer, the AIBO robots try to recover their exact position in the field by using different techniques. First vision-oriented routines are run to extract useful data of the robot’s view. After that step, a second set of procedures based on sound source localization is performed. The goal of the sound localization component is to determine the azimuth of the observed sound source. The procedure consists in finding the horizontal direction expressed as the angular distance between the robot’s orientation and the direction of the sound source. The azimuth is determined by using the interaural time delay (ITD). ITD stands for the difference in arrival time of the sound source's signal between the left ear and the right ear. The sound source localization is done only on a number of separate, predefined frequencies so as to be able to discern multiple sound sources from one another and facilitate the distinction between signal and noise.

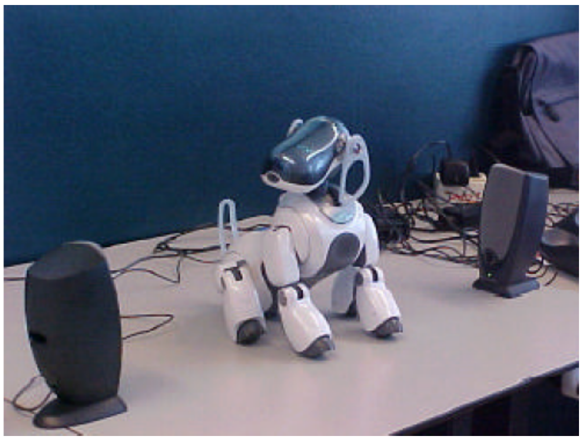

**Figure 1. AIBO robot detecting sound sources**

### Signal Reliability

A threshold-oriented mechanism is set to prevent sound source localization from being performed on frequencies that contain only background noise. The reliability mechanism includes a Fast Fourier Transform on both the left and right channel of the sampled signal. The reliability threshold $t$ is defined as in Formula 1.

$$t = \mu + k \cdot \left( \frac{|f_i^l| + |f_i^r|}{2} - \mu \right)^2 \quad (1)$$

Parameter $\mu$ is defined as in Formula 2.

$$\mu = \operatorname*{mean}_{\forall i}\left( |f_i^l| + |f_i^r| \right) \quad (2)$$

The elements in Formula 3 represent the amplitudes of the $i$-th Fourier components of the left and right signal, respectively.

$$|f_i^l| \text{ and } |f_i^r| \quad (3)$$

$k$ is a constant which for has a value of 2, so that the sound from the ERS–7's speaker does not raise the threshold too much, which would cause it to view the signals of other ERS–7s as unreliable. The signal corresponding to the $j$-th Fourier component is then considered reliable if and only if the relation in Formula 4 takes places.

$$\left( |f_j^l| > t \right) \vee \left( |f_j^r| > t \right) \quad (4)$$

Performing the **FFT** on a sampling frame of 16 ms, the *j*-th Fourier component corresponds with a frequency of *j* / 0.016 = *j* · 62.5 Hz.

## Calculating the ITD

Calculating the **ITD** may imply a lot of computation. The fastest way is to perform a **FFT** on the samples signal and then use that to determine the interaural phase difference. This method is extremely sensitive to reflection and background noise and it was found to be of no practical use in a real world application like the soccer or rescue game. In order to overcome the limitation, a slightly complex algorithm was developed to calculate the **ITD**.

The first stage of processing on the developed algorithm specifies that both, the left and the right signal are run through a filter bank to extract the signals at those frequencies in which there is any interest. The filter bank is filled with bandpass filter of which each central frequency is a multiple of 250 Hz. Every filter in the filter bank is a 128th-order **FIR** filter with a bandwidth of 62.5 Hz, designed using a 129-point Bartlett window. One beneficial property of this specific type of filter is that it has a magnitude response of −45 dB or less for all frequencies that are multiples of 250 Hz, other than the central frequency (Figure 2).

The next step is to determine which **ITD** maximizes the cross-correlation between the left and the right signal, for each frequency.

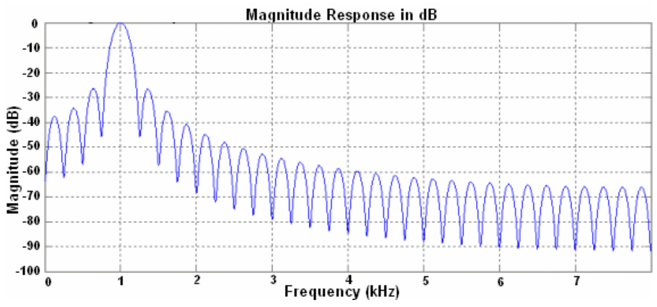

**Figure 2. Magnitude Response of Band-pass Filter with 1000 Hz Central Frequency**

In the windowed cross-correlation defined in Formula 5 $l[n]$ and $r[n]$ are the discretely-filtered left and right signal, respectively, and where $N_1$ and $N_2$ define a window in time for which the cross-correlation is calculated.

$$\rho_{lr}[d] = \sum_{n=N_1}^{N_2} l[n]r[n-d] \qquad (5)$$

The value of the delay $d$ that maximizes $\rho_{lr}[d]$ is the **ITD** in samples. This is then multiplied by the duration of one sample to acquire the **ITD** in seconds.

A problem with this cross-correlation method is that for signals of constant amplitude it produces more than one maximum at intervals equal to the signal's frequency.

In order to overcome the ambiguity, an amplitude-modulated signal as the tested sound source was analyzed. More specifically, a triangular modulation was performed since it always produces the greatest difference in amplitude between periods.

Furthermore, when the **ITD** is larger than one half the period of the signal, the smallest delay which maximizes the cross-correlation is actually smaller than the real **ITD**.

To avoid this issue of wrapping, the signals of frequencies that have wavelengths shorter than half the distance between both ears are not processed. The final model described in the next section provides clear minimum and maximum **ITD** values.

## The Azimuth Model

Determining the azimuth from the observed **ITD** requires a model that predicts the difference in distance between the left and right ear that the signal of a sound source on a specific azimuth must travel in order to reach them, respectively.
As testing, the Lord Rayleigh's (John Strutt's) duplex theory has been tested initially. However, the real world measures with the AIBO (ERS–7) robot have definitely shown with great certainty that it is possible for **ITD**s to occur that are greater than can be predicted with this model. Therefore, it has been decided to design and use a new model.

A major problem with Lord Rayleigh's model is that he assumes that the left and right signals arrive at the head in two parallel lines. Unfortunately, this is only the case if the sound source is of the exact same size as the head. In the testing environment, the robot’s own speaker has been used (while another AIBO ERS–7 fulfills the role of observer). The speaker is significantly smaller than the head. Because of this, the paths of the left and right signal are at an angle to each other. This phenomenon becomes even stronger the closer the sound source is positioned to head.

The used model represents the sound source as a point source, since this is the most encountered situation.

$$ITD = \frac{v_s}{r_m}\left((2-\chi)\theta + \chi\sin(\theta)\right) \qquad (6)$$

In the approximation given as in Formula 6, $v_s$ equals the speed of sound, $r_m$ equals microphone radius (the distance of either ear to the center of the head), $\theta$ is the azimuth, and $\chi$ is a function of the distance of the sound source to the head. The properties of $\chi$ are that it is always positive and correlates with the distance of the sound source to the head, and that as the sound source moves toward the head, $\chi$ converges to zero, while as the distance of the sound source to the head grows toward infinity, $\chi$ converges to 1 (which leads to Lord Rayleigh's formula).

For computational simplicity in converting the ITD to the azimuth, it is assumed that the distance of the sound source to the head is effectively zero, so that $\chi$ equals zero (Formula 7).

$$\theta = \frac{r_m}{2v_s} ITD \quad (7)$$

Since the distance of the sound source to the head will in reality never be zero, any $\theta$ would seem to lead to a slight overestimation of the ITD in this model, and conversely, any ITD would lead to an underestimation of the $\theta$. In practice, this bias is only noticeable when the sound source is located a significant distance above the azimuth plane or when the distance of the sound source to the head becomes larger than roughly three meters.
Furthermore, since the speaker of the AIBO robot, which serves as the sound source, is located significantly closer to the ground than its ears, interposing parts of the observing ERS–7's body (mainly the shoulders and nose) lead to additional delays which even further obscure this bias.

## Deconfusion

By using the mechanisms previously described, the ITD has been converted to an azimuth. At the current processing stage, the azimuth is *confuse* by not being clear whether it is really a *front angle* or merely a mirrored *back angle* (Figure 3).
Indeed, the method described above can **only** extract front angles—azimuths of absolute value equal to or smaller than $\pi/2$. All back angles—azimuths of absolute value between $\pi/2$ and $\pi$—are merely mirror along the axis of the head that runs through $-\pi/2$ and $\pi/2$. In order to discover which azimuths are front angles and which are not, a *deconfusion* process has to be run.

The current *deconfusion* approach operates on some given premises. For a certain value of the head rotation $\delta$:

- If a sound source at a certain azimuth $\theta_{old}$ was previously localized and the heading was rotated **toward** it:
  - If $-\pi/2 + \theta_{old} < \delta < -\pi/2 - \theta_{old}$ or $\pi/2 - \theta_{old} < \delta < \pi/2 + \theta_{old}$, then the sound source must be in **front** of the heading.
  - If $|\theta_{new}| > |\theta_{old}|$, then the sound source must be **behind** the heading.
- If a sound source at a certain azimuth $\theta_{old}$ was previously localized and the heading was rotated **away** from it:
  - If $-\pi/2 - \theta_{old} < \delta < -\pi/2 + \theta_{old}$ or $\pi/2 + \theta_{old} < \delta < \pi/2 - \theta_{old}$, then the sound source must be **behind** the heading.
  - If $|\theta_{new}| > |\theta_{old}|$, then the sound source must be in **front** of the heading.
- Otherwise, it is assumed per default that the sound source is in **front** of the heading.

If the rotation of the head is not too far, wrapping can occur which can cause the deconfusion to fail. In practice, the case is avoided by not allowing the robot to rotate the head more than $\pm\pi/2$ between observations, which is a safe value for any azimuth.

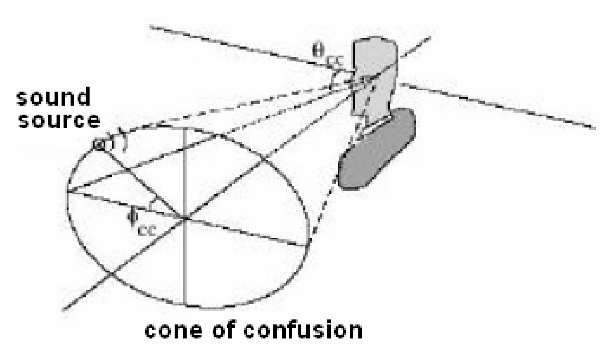

**Figure 3. Cone of confusion**

## Test results on source sound detection

At first, SoundSourceLocalizer was designed as a passive Object. It would only receive data from the AIBO and not control any parts of the robot itself. The necessary movement of the head for deconfusion would have to be done by other applications. After testing however, we noticed that the calculated azimuths were fairly inaccurate. This could be attributed to the sound produced by moving the head while sampling. It was decided that we would also add an active mode of operation to SoundSourceLocalizer, in addition to the already existing passive mode. In the active mode SoundSourceLocalizer would listen for a sound, move the head towards the perceived source using the Object HeadMover, listen again, and repeat.
The defined goal in practical testing for the playing context assumes the possibility that the robot can make the head rotations while it is listening for sound sources at the same time. Unfortunately, doing so deforms the signal in such a way that it seriously skews both ITD calculation and deconfusion.
Therefore, some rules had to be set so as to always completely stop the robot's head movement to safely listen for sound sources (recording azimuths from 10 sampling frames) and only then moves its head to respond, after which the cycle is repeated, *ad infinitum*.

Obviously, if the head is not rotated between two periods of observations, no deconfusion is possible. In those cases were the AIBO robot would not move the head enough between observation periods to allow for deconfusion, it is forced to make *saccadic* head movements. These relatively small but significantly large enough motions are randomly distributed and allow our software to continue to successfully perform deconfusion at all times.

In trying to get a better estimate of the true azimuth of a sound source, a smoothed estimation of the azimuth over time was imposed (Formula 8)

$$\theta_{n+1} = (1-\alpha)\theta_n + \theta \qquad (8)$$

The $\theta$ is the latest observed, deconfused azimuth, and $\theta_n$ is the smoothed azimuth at the *n*-th sampling frame. For the implementation of the algorithm, the value of $\alpha$ was chosen to be equal to 0.1, which gives the last 10 sampling frames an influence of about 65% on the current value of the smoothed azimuth, as can be seen from Figure 2.
This makes for less fluctuation of azimuth values and also greatly reduces the influence of incorrect azimuths (due to background noise, etc.).

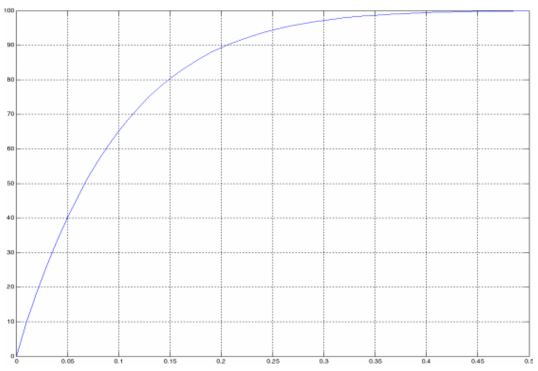

**Figure 4. Influence of Last 10 Frames on Smoothed Azimuth**

Another drawback of using cross-correlation is that its accuracy greatly depends on the sampling rate. That is because the ITD can never be calculated more accurately than the length in seconds of a sample. Fortunately, the random nature of the saccadic head movements partially alleviates this. When the azimuth is smoothed over time, the shaking motions of the AIBO robot slowly cause it to converge to a more accurate estimate.

## PLAYING RESCUE GAME

An additional task is to set the AIBO robots to play in the Rescue game. The requirements that exist at the low functional level include the presence of the specific hardware sensors to be connected to each playing robot. The more sensors are available, the more information is accessible for creating the global image on the existent crisis events in the scene. Given the data related to the environmental events and the long-time decision previously set, the robot has to take local temporal actions for approaching the target. The robot can also receive useful information about emergency events taking place in other locations associated to the current game session. Each robot has detection rules concerning safety and emergency management activities and possible actions to be taken for each known context. The high-risk activities in the context of the game include those activities associated with accident and emergency management, household, water, swimming, boating, fire, marine, environmental, vehicle use and other activities.

## VISION ORIENTATION

In addition to the sound-processing module that is used for determining the location of the objects that emit sound in the scene, the vision module follows a set of commands for extracting the useful graphic information. The tasks consist in recovering the relative position of the ball, the field lines, the goals, the flags, the other players, and the obstacles.
For the detection of the ball, the image is searched for candidate positions for the ball. The essential criterion for image analysis stands for color and shape-oriented detection. The information concerning the positions and angles of the detected objects are stored relative to the robot.

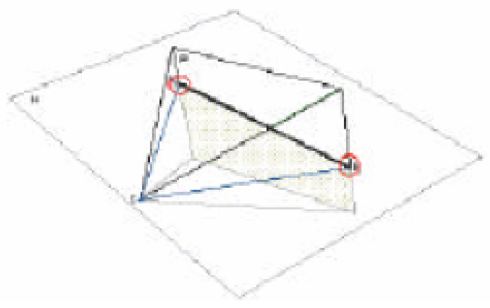

**Figure 5. Virtual projection of the camera matrix against the ground plane**

The most important feature for detection is the horizon.
It is the line that describes the angle of the head relative to the ground (Figure 5). The two red circles are the start and end points of the horizon.

### Ball detection

For balls, upper and lower points on their boundaries are detected during scanning. Points on the border of the image are ignored. During scanning, red pixels below a reasonable number of orange pixels are also treated as orange pixels, because shaded orange often appears as red. Although only a single ball exists in the game, the points are clustered before the actual ball position is detected, because some of them may be outliers for a red robot. To remove outliers in vertical direction, upper points are ignored if they are below many other lower points in the same cluster, and lower points are ignored if they are above many other upper points in the same cluster.
The actual calculation of the position of the ball depends on the number of points detected. If at least three points have been detected, these can be used to calculate the centre of the ball by intersecting the middle perpendiculars (Figure 6). So, three points are selected to construct two middle perpendiculars.
Considering that the three small red circles are the found ballpoints, the dark grey lines are the middle perpendiculars and the large red circle the assumed centre of the ball. From that point, the distance to any ballpoint can be taken as the radius of the circle.

Points that are close to green are preferred, because it is assumed that they are actually located on the border of the ball. In contrast, points close to white can result from a high spot on the ball, but also from the real border of a ball in front of the border of the field. First, two point are selected that have the largest distance from each other. Then a third point is chosen that is furthest away from the two other points.

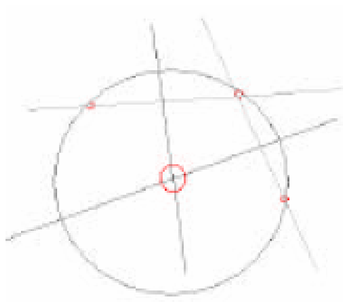

**Figure 6. The recovering of the ball geometry**

If the three points do not lie on a straight line, the centre of the ball in the image can be calculated, even if the ball is only partially visible. If the ball is not below the horizon or if the camera matrix is not valid because the robot is currently kicking, the distance to the ball is determined from its radius. Otherwise, the distance is determined from the intersection of the ray that starts in the camera and points to the centre of the ball with a plane that is parallel to the field, but on the height of the ball centre.
If there is at least a single point on the border of the ball, it is assumed to either be the highest or the lowest point of the ball. The point is projected to the field plane and the distance to the ball is determined from this projection. If all orange points lie on the border of the image, it is assumed that the ball fills the whole image and the middle between all orange border points is assumed to be centre of the ball. The position on the field is again determined by intersecting the view ray with the field plane in the height of the ball centre. Finally, the position of the ball in field coordinates is projected back into the image, and a disk around this position is sampled for orange pixels. If enough orange pixels are found, t he ball is assumed to be valid.

## Flag (field corner) detection

All indications for flags found during scanning the grid are clustered. In each cluster there can actually be indications for different flags, but only if one flag got more indications than the others, it is actually used. The centre of a cluster is used as a starting point for the *flag specialist*. It measures the height and the width of a flag. From the initialization pixel, the image is scanned for the border of the flag to the top, right, down, and left where top/down means perpendicular to the horizon and left/right means parallel to the horizon. This leads to a first approximation of the size of the flag. Two more horizontal lines are scanned in the pink part and if the flag has a yellow or a sky-blue part, two more horizontal lines are also scanned there. The width of the green part of the pink/green flags is not used, because it is not always possible to distinguish it from the background. To determine the height of the flag, three additional vertical lines are scanned. The leftmost, rightmost, topmost, and lowest points found by these scans determine the size of the flag. To find the border of a flag, the flag specialist searches the last pixel having one of the colors of the current flag. Smaller gaps with no color are accepted. This requires the color table to be very accurate for pink, yellow, and sky-blue.

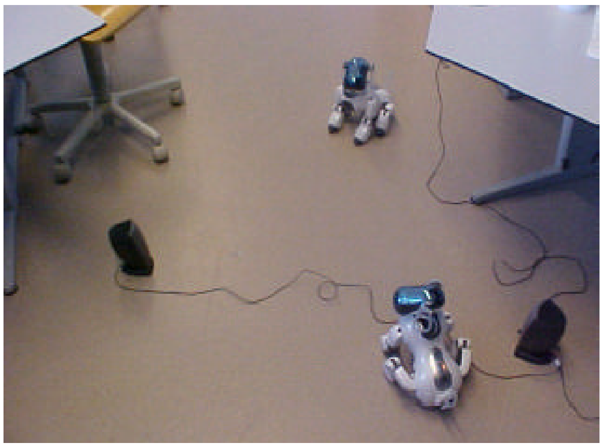

**Figure 7. Playing AIBOs in speaker environment**

### Goal detection

A goal *specialist* measures the height and the width of a goal. The image is scanned for the borders of the goal from the left to the right and from the top bottom, where again top/down means perpendicular to the horizon and left/right parallel to the horizon. To find the border of the goal the specialist searches the last pixel having the color of the goal.
Smaller gaps with unclassified color are accepted. The maximal size in each direction determines the size of the goal.

### Robot detection

To determine the indications for other robots, the scan lines are searched for the colors of the tricots of the robots. If a reasonably number of pixels with such a color is found on a scan line, it is distinguished between two cases. If the number of pixels in tricot color found on a scan line is above a certain threshold, it is assumed that the other robot is close. In that case, the upper border of its tricot (ignoring the head) is used to determine the distance to that robot. This is achieved by intersecting the view ray through this pixel with a plane that is parallel to the field, but on the “typical” height of a robot tricot. As the other robot is close, a misjudgment of the “typical” tricot height does not change the result of the distance calculation very much. As a result, the distance to close robots can be determined.
If the number of pixels in tricot color found is smaller, it is assumed that the other robot is further away. In that case, the scan lines are followed until the green of the field appears. Thus the foot points of

the robot are detected. From these foot points, the distance to the robot can be determined by intersecting the view ray with the field plane. As not all foot points will actually be below the robot's feet (some will be below the body), they are clustered and the smallest distance is used.

## CONCLUSION

The multimodal capabilities present great advantages for robot coordination in playing games. The fusion of information is done at distinct processing levels so as to obtain useful environmental details. A research of a great importance is to combine results from the processing on different type of signal data so as to establish the actions for the next period of time. In the case of AIBO robot there are some modules that manage the flow of data from the internal sensors.

In the current paper we tried to research different ways of combining information taken as results of the supported processing modules so as to obtain the best performance for the robot playing in games.

A decisive criterion was to evaluate the individual achievements of the robot and also the performance of the team. We also found algorithms for low-level processing of the signal from the sensors. For the sound analysis, the AIBO can sit listening to one particular frequency and move its head towards the sound source, or else indicate that the sound source is beyond the turning range of its head. Other sounds, nearby objects, positioning of the sound source behind the head or talking people did not seem to have a significant influence on AIBO's ability to localize the sound source.

Observing the performance of the final software, we can conclude that sound source localization is very possible on an embedded environment such as AIBO ERS-7.

It has enough system resources to process the necessary operations and calculations fast enough to follow a sound source in real-time. Sound source localization can be made fairly accurate and reliable to the point where two AIBOs can blindly find each other by using sound. The Dutch AIBO Team demonstrated this during the Robocup Tournament 2004, in a demo using this sound source localization software.

For the vision part we designed algorithms for recovering position of some important objects such as the ball, the field lines, the goals, the flags, the other players, and the obstacles.

## REFERENCES

D.Datcu, L.J.M.Rothkrantz: 'A multimodal workbench for automatic surveillance', Euromedia Conference, Hasselt, 2004.

D.L.Hall, J.Llinas: 'Handbook of multisensor data fusion' CRC Press, 2001.

H.J.W.Spoelder, D.M. Germans, L. Renambot, H.E. Bal, P.J. de Waal, and F.C.A. Groen: 'A framework for interaction of distributed autonomous systems and human supervisors', IEEE Transactions on Instrumentation and Measurement, *51*(4):798-803, August 2002.

J.A.Larson, T.V.Raman: 'W3C multimodal interaction framework', http://www.w3.org/TR/mmi-framework, W3C Note. December 2002.

J. P. Jansen: 'Looking like humans do. An approach to robust robot vision', T.U.Delft report MMI-2004 - 11.

M.Pantic and L.J.M.Rothkrantz: 'Towards an affect sensitive Multimodal HCI', Proceedings of the IEEE, Special Issue on Multimodal Human-Computer Interaction (HCI), vol 91, no. 4, 1370-1390, 2003.

M.Pantic, L.J.M.Rothkrantz: 'Automatic analysis of facial expressions: The State of the art', IEEE transactions on Pattern Analysis and Machine Intelligence 22(12): 1424-1445, 2000.

M.Pantic, L.J.M.Rothkrantz: 'Expert system for automatic analysis of facial expression', Image and Vision Computing 18/2000, 881-905.

M. Richert, T. Roberti, W. de Vries: 'Sound source localization using the AIBO ERS-7', T.U.Delft report MMI-2004-10.

N.Krahnstoever, S.Kettebekov, M.Yeasin, and R.Sharma: 'A real-time framework for natural multimodal interaction with large screen displays' In Proc. of Fourth Intl. Conference on Multimodal Interfaces (ICMI 2002), Pittsburgh, PA, USA,October 2002.

AIBO Entertainment Robot
http://www.us.aibo.com/, http://openr.aibo.com