J Intell Robot Syst (2009) 55:81–106
DOI 10.1007/s10846-008-9291-9

UNMANNED SYSTEMS PAPER

Advanced Adaptive Sonar for Mapping Applications

A. Hernández · J. Ureña · M. Mazo · J. J. García ·
A. Jiménez · J. A. Jiménez · M. C. Pérez · F. J. Álvarez ·
C. De Marziani · J. P. Dérutin · J. Sérot

Received: 6 December 2007 / Accepted: 24 September 2008 / Published online: 18 October 2008
© Springer Science + Business Media B.V. 2008

Abstract An advanced adaptive sonar module is described, capable of being configured to different circumstances and distances according to reflectors found in the environment. Thanks to the sensory distribution, it is possible to identify three basic types of reflector (planes, edges and corners). Furthermore, a heuristic map of the environment is built. The proposed methods have been computationally optimized, and implemented in a real-time system based on a Field-Programmable Gate Array (FPGA) and a Digital Signal Processor (DSP). Results have been obtained in the detection, classification and mapping of obstacles; and finally testing has been carried out on a commercial vehicle.

Keywords Ultrasonic sensory system · Mapping applications · Real-time implementation

A. Hernández (✉) · J. Ureña · M. Mazo · J. J. García · A. Jiménez ·
J. A. Jiménez · M. C. Pérez
Electronics Department, University of Alcalá,
28806, Alcalá de Henares, Spain
e-mail: alvaro@depeca.uah.es

F. J. Álvarez
Department of Electrical Engineering,
Electronics and Automatics, University of Extremadura,
06071, Badajoz, Spain

C. De Marziani
Electronics Department, National University of the Patagonia San Juan Bosco,
9005, Comodoro Rivadavia, Argentina

J. P. Dérutin · J. Sérot
LASMEA Laboratory (UMR 6602 CNRS), Blaise Pascal University,
63177, Aubière Cedex, France

1 Introduction

Traditionally, ultrasonic transducers have been used in several robotic applications to perform different tasks such as obstacle avoidance [1–4], mobile robot guidance [5, 6], or building of environment maps [7, 8]. Sometimes, only distances are measured between reflectors and transducers, so a reduced amount of information is provided. In other cases, the ultrasonic signal is processed in a more complex way, collecting as much information as possible from the environment. One of the common solutions for the low-level ultrasonic signal processing is to obtain time-of-flight (TOF) [9, 10]. TOFs determine only a distance between the reflector and the transducer, whenever the transducer works as an emitter and receiver. Nevertheless, more information can be obtained from these TOFs: high-level analysis is composed by those algorithms, that carry out a more complex interpretation of the TOFs measured by several transducers [8, 11–19].

In order to make possible the use of TOFs in high-level tasks, high precision is required in the measurements. Different techniques provide the desired precision, being the encoding of the ultrasonic emission one of the most usual. Regarding this encoding, different sequences have been successfully tested: Barker code [20–22], pseudo-random sequences [5], or Golay sequence complementary sets [23, 24].

The association of ultrasonic transducers in more complex sensory arrays usually allows the identification of obstacles (planes, edges and corners), as well as the determination of the lateral resolution [6, 10, 17–19, 25–28]. One of the possible purposes of a sonar system installed on a mobile robot is to provide a map of the environment [12, 14] with enough quality to solve navigation, localization or similar applications. Building of maps is often based on the detection and identification of different obstacles (reflectors). Two possible choices can be used for mapping: SLAM algorithms [12, 14, 16, 17, 29] and statistical grids [8, 11, 13, 21, 30, 31]. On one hand, SLAM algorithms are based on a feature map built according to the distances, dimensions and types of the detected reflectors. On the other hand, in grid-based methods the environment is divided into a Cartesian representation, with the number of cells depending on the measurement precision required and the computing resources available.

Ultrasonic sensory systems provide some advantages and drawbacks, compared to other types of sensors. Those based on laser range-finder often have high costs and low exploration speeds; furthermore, they become unreliable under certain light conditions or on interaction with some materials (e.g. glass) [32, 33]. On the other hand, vision systems present high computational load, as well as difficulties in the extraction and tracking of objects in a stable way [34]. Nevertheless, ultrasonic systems are usually low cost, and they do not require so high computational loads; they also provide a reliable operation, with reduced dependencies on environmental conditions, and suitable scanning speeds. The main drawback is the detection probability of objects, decreased by specular reflections in some cases, and multiple reflections, that can provide phantom objects in wrong positions (artifacts). These problems though can be mitigated with a suitable configuration of transducers in an array.

Regarding algorithm implementation, initial approaches were based on programmable systems [26, 35]. Nevertheless, the improvement in the features of ultrasonic systems has led to more complex algorithms, where real-time operation has become more difficult to achieve. In this way, FPGA- and DSP-based systems have recently provided suitable proposals [36–38] for the real-time implementation of ultrasonic signal processing.

This work presents a novel and efficient implementation of an advanced sonar processing, comprising an accurate TOF determination and a classification and/or mapping application. Section 2 describes the use of sensory distribution, and how TOFs are determined in an optimal way. Section 3 deals with the high-level proposal. The integration of high- and low-level algorithms is explained in Section 4, whereas the temporal distribution and performance is detailed in Section 5. Section 6 shows the obtained results, and finally some conclusions are discussed in Section 7.

2 Determination of Time-of-Flight

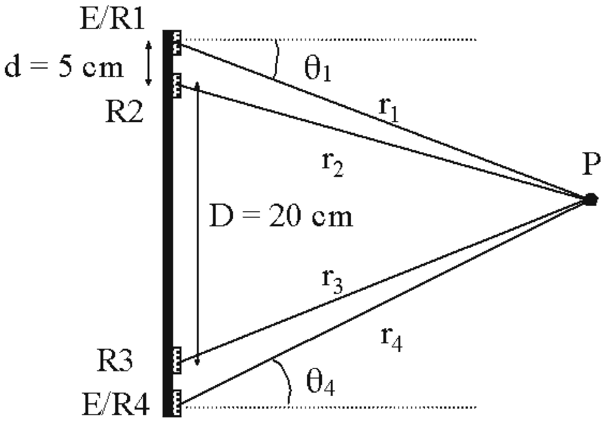
2.1 Structure of the Sonar Module

Figure 1 shows the configuration of the used ultrasonic sensor, formed by four transducers gathered in two vectors, and based on the definition carried out in [21]. The two transducers located at each end constitute a vector, which allows the measurement of distances and reception angles—lateral resolution. The two transducers of a vector are close one to each other, thus minimizing the correspondence problem between echoes, and allowing the reception of echoes coming from the same reflector in both transducers. Figure 2 depicts the physical aspect of the mentioned sonar module.

The external transducers, E/R1 and E/R4, are the only ones capable of emission, whereas all of them are capable of receiving echoes from the environment. The parameters r_i, $i = 1, 2, 3, 4$, represent the distances between every ultrasonic transducer and a reflector P located in front of the system. These distances determine the value of the detected TOFs.

The simultaneous reception allows to have four TOFs, increasing the probability of the obstacle detection, and reducing problems associated with specular reflections. By assigning to each emitter an unequivocal sequence (orthogonal to the others), it

Fig. 1 Geometrical distribution of the ultrasonic sensor used



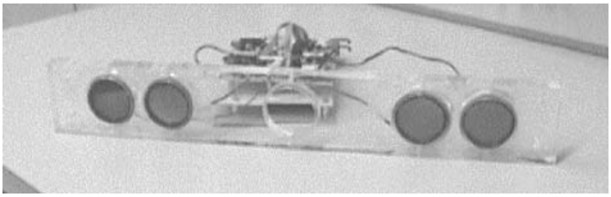


Fig. 2 Global view of the developed module

is possible to discriminate the source of the echoes detected by any transducer. This implies a reduction of the scanning time, by means of the simultaneous transmission in the emitters, and it allows TOF measurements with moving vehicles or robots [23]. In this way, with sequential non-simultaneous emissions, robots often are stopped and the environment should be static during successive emissions to keep the same geometry for all the emitters. Furthermore, it is also useful in reflector classification, since it is necessary to process echoes caused by the same reflector and coming from different emitters [26]. On the other hand, this setup does not provide more unwanted receptions in comparison with sequential proposals, since only the first echo is considered in each transducer, so other reflections with longer paths are not processed by the system.

2.2 Encoding of the Ultrasonic Emission

The use of Golay complementary pairs to encode the ultrasonic emission and determine the TOFs has been verified in previous works [23, 39]. Compared to other proposals, as those based on Barker codes [20, 21] or pseudo-random sequences [5], complementary sets of sequences usually provide suitable auto- and cross-correlation features, with the possibility of changing their lengths according to the system requirements. Furthermore, Golay complementary pairs present efficient correlation structures, where the computational load is significantly reduced. In this way, although not only Golay pairs but also other codes as pseudo-random provide similar features encoding, the reduction in the computational load makes Golay pairs suitable in real-time implementations.

The Golay pair is composed of two sequences, $A[n]$ and $B[n]$, where the addition of both independent auto-correlation functions provides an ideal signal according to Eq. 1.

$$C_{AA}[n] + C_{BB}[n] = \begin{cases} 2N & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{1}$$

Where $A[n]$ and $B[n]$ are sequences with values in $\{-1, +1\}$; N is the number of bits, or the length of sequences; and $C_{AA}[n]$ and $C_{BB}[n]$ are the auto-correlation functions of both sequences, $A[n]$ and $B[n]$ respectively. Figure 3 shows the result of this process for a 32-bit Golay complementary pair.

For the simultaneous emission of the pair, a digital variant of a classic QPSK (Quadrature Phase Shift Keying) modulation has been implemented, so the components I and Q of the modulation has been associated to each one of the sequences

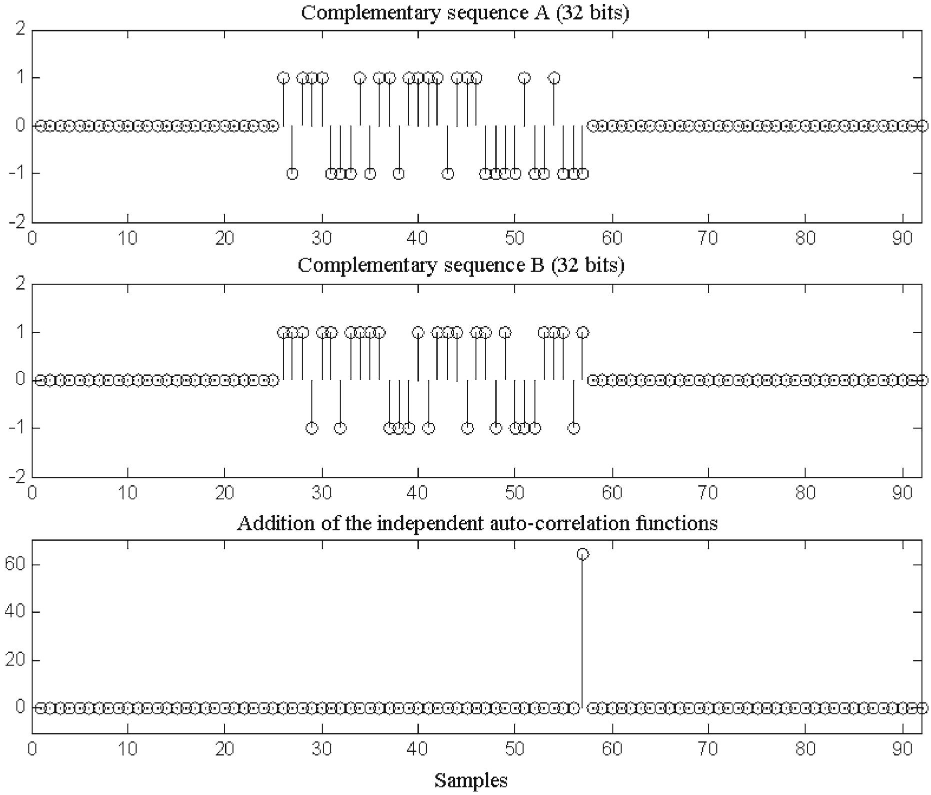


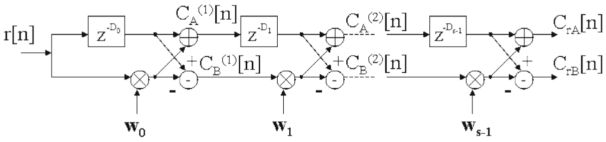
Fig. 3 Example of an auto-correlation function for a 32-bit Golay complementary pair

of the pair [23]. The modulation carrier presents a frequency of 50 kHz to center the emission at the maximum spectral response of the transducer.

On the other hand, the correlation in the reception stage can be considered simplified with the use of the Efficient Golay Correlator (EGC) [40, 41]. This allows the reduction of computational load and the simplification of the detection process, whenever Golay sequences with a length N power of 2 are used ($N = 2^s$, where s is the number of bits of the sequence seed $\mathbf{W}$, such that $\mathbf{W} = [w_0, w_1, \ldots w_{s-1}]$). The block diagram in Fig. 4 shows this efficient correlator.

Where D_s means a delay module $D_s = 2^{P_s}$; P_s is any permutation of the numbers 0, 1,..., s-1; $C_{rA}[n]$ and $C_{rB}[n]$ are the results of the correlation between the input signal $r[n]$ and the pair of Golay sequences A and B, generated by using the seed $\mathbf{W} = [w_0, w_1, \ldots w_{s-1}]$.

Fig. 4 Block diagram of the Efficient Golay Correlator (EGC)



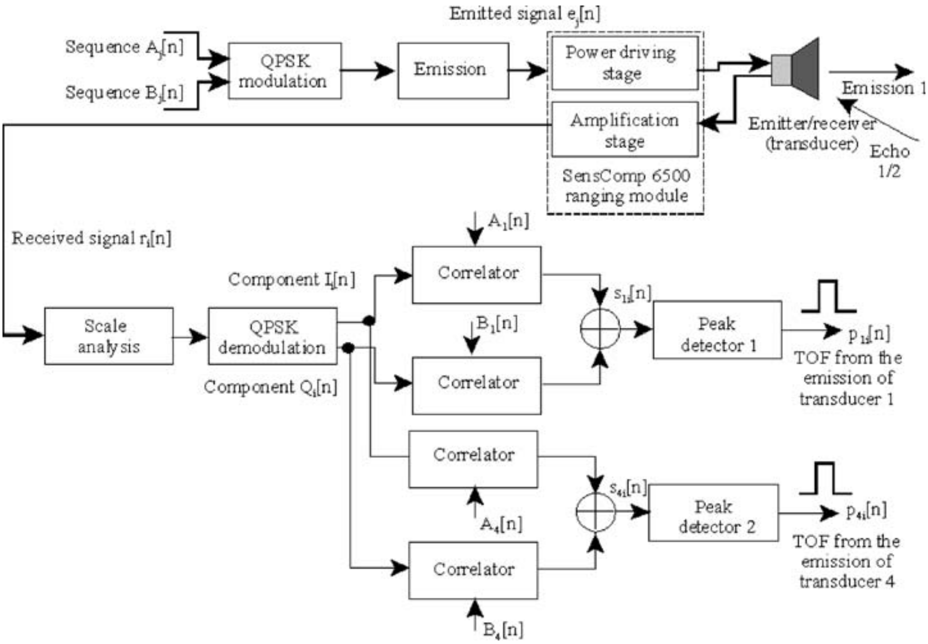


Fig. 5 Simplified scheme of the processing system in an emitter/receiver

Furthermore, the echo detection processing can be generalized at every receptor for the detection of two different and orthogonal Golay complementary pairs, which have been previously QPSK modulated (see the block diagram for an emitter/receiver in Fig. 5). If the block diagram for only one receiver (not emitter) is considered, the modules related to the emitting process are not needed (modulation and emission modules in Fig. 5). Furthermore, a dataflow graph has been defined (see Fig. 6) to show the execution flow of the different tasks in this low-level processing, as well as the possible data dependencies existing among them. Note that H_i is the modulation of the Golay pair; N_i is the emission of the modulated signal (with a carrier frequency of 50 kHz); Q_i is the acquisition of a new sample from the received signal (assuming a sampling frequency of $f_S = 400$ kHz, every new sample should be processed in $T_S = 2.5$ μs); F_i is a preamplifying operation; D_i is the demodulation; G_{ji} is the Golay pair correlation; and P_{ji} is the peak detection ($j = \{1, 4\}$ is the emitter index and $i = \{1, 2, 3, 4\}$ is the receiver index).

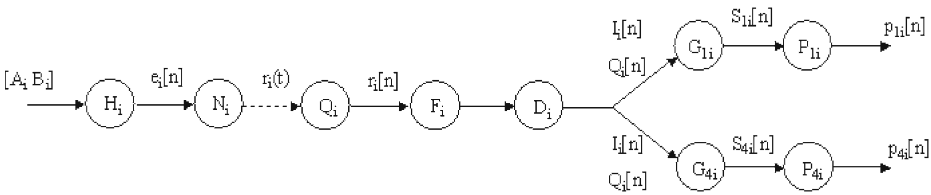


Fig. 6 Graphical representation of the low-level processing dataflow

3 Building Maps of the Environment

As explained before, the results from low-level processing are the TOFs. The system, formed by four transducers, is capable to provide eight TOFs, after s single emission process. Starting from those TOFs, the application presented here obtains a map of the scanned environment. The proposed algorithm has a characteristic operation mode, which allows the adaptation of the system to the scanned environment. Based on the distances measured by TOFs, the following different reception areas are defined:

- Near zone. The system is in this mode whenever distances between reflectors and transducers are lower than the distance $D_{\mathrm{NM}} = 100$ cm. The 32-bit Golay complementary sequences are used, in order to reduce the emitted power. For this zone, cells in the map have a resolution of 10 cm × 10 cm.
- Middle zone. The benefits of the system using 32-bit sequences remarkably diminish when the detected reflector moves far away from the system, since a worse signal-to-noise ratio (SNR) is obtained. The sequence used up to the distance D_{MF} has a length $N = 64$ bits, where D_{MF} is the limit between the middle zone and the far zone explained next (fixed at 300 cm). The cells still keep the dimension 10 cm × 10 cm.
- Far zone. Here reflectors are further than a distance $D_{\mathrm{MF}} = 300$ cm. Due to the increase in distance, 128-bit Golay sequences are used to work with more reduced SNR. Also, since results become less accurate, the cell dimension is fixed at 20 cm × 20 cm.

Note that the dimensions given to cells are not constrained by the precision of the ultrasonic sensor, but by the computational load of the real-time implementation: as will be explained afterwards, the resources and the device utilization limit some features of the system. Figure 7 shows a spatial representation of these three operation modes.

3.1 Algorithms for the Near and Middle Zone

The starting point of this processing is the four TOFs $[t_{j1}\ t_{j2}\ t_{j3}\ t_{j4}]$ obtained after the emission carried out by an emitter j. Since there are two transducers with emission capacity, E/R1 and E/R4, the procedure explained next should be executed twice. The processing in the near and middle zone can be divided into the following steps (note that the nomenclature assigned to each task has been enclosed in brackets):

- Correspondence analysis (tasks A_{jA}, A_{jB} and A_{jS}).
  The correspondence analysis tasks (A_{jA} and A_{jB} correspond to vectors A and B, and A_{jS} for the global sensor) tries to determine if, for every emission j, the echoes received by the transducers come from the same reflector. This is the first step before classification. For each emission j, this study is performed in both vectors, A and B (A_{jA}, A_{jB}), and in the global sensor (A_{jS}) [22]. In vectors, the echo correspondence is confirmed by Eq. 2.

$$|t_{jm} - t_{jn}| \leq \frac{d}{c} \cdot \sin\frac{\alpha}{2} \tag{2}$$

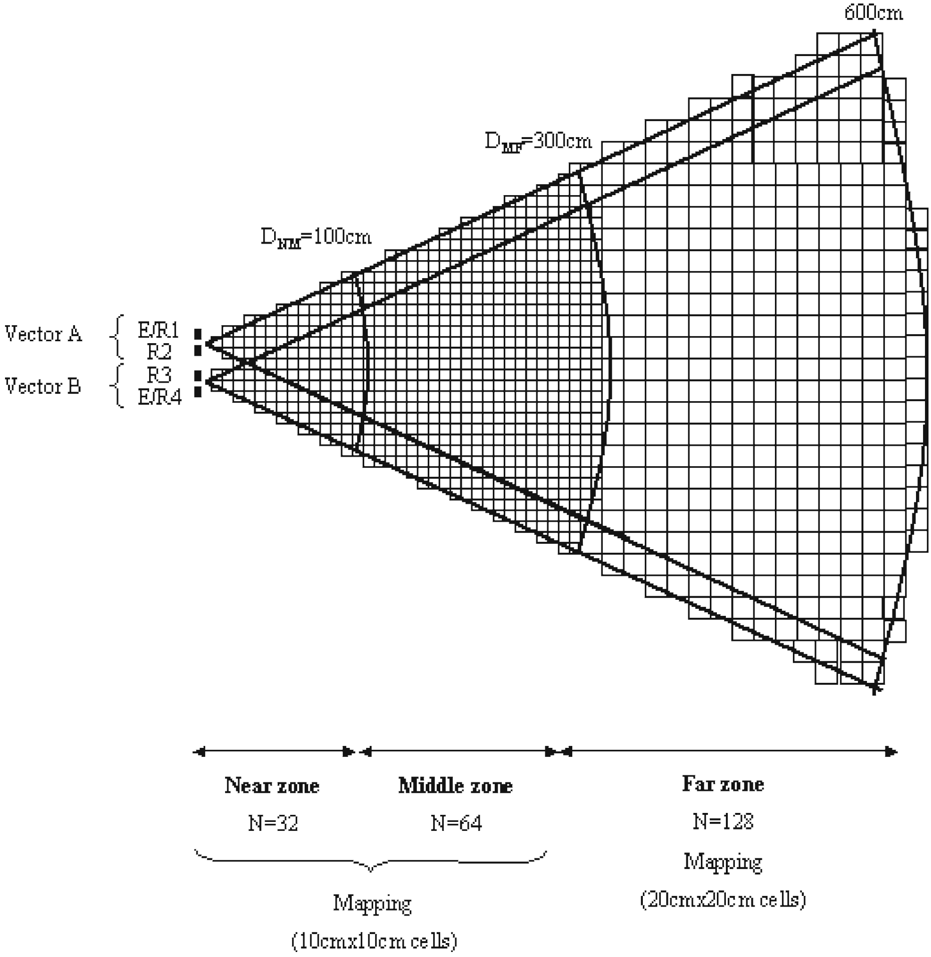


Fig. 7 Representation of the operation modes in the proposed algorithm

Where m and n are the two transducers of a vector, $m, n = \{1, 2, 3, 4\}$; thus t_{jm} and t_{jn} are the TOFs measured by each transducer for the emission j; d is the distance between them; c is the speed of ultrasound in air; and α is the aperture angle of transducer [42]. Similarly, there exists echo correspondence in the global sensor if TOFs meet Eq. 4.

$$t_{jA} = \min[t_{j1}, t_{j2}] \qquad t_{jB} = \min[t_{j3}, t_{j4}] \tag{3}$$

$$|t_{jA} - t_{jB}| \leq \frac{D + d}{c} \cdot \sin\frac{\alpha}{2} \tag{4}$$

Where t_{j1}, t_{j2}, t_{j3} and t_{j4} are the TOFs determined in each transducer for the emission j; t_{jA} and t_{jB} are the minimum TOFs in both vectors; and D is the distance between vector A and B.

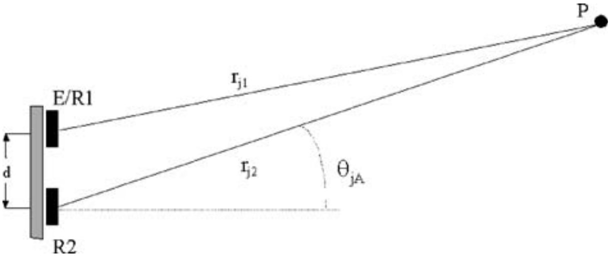


Fig. 8 Detection of the reflector P by the vector A

- Determination of distances r_{jA} and r_{jB} between vectors and reflectors (tasks R_{jA} and R_{jB}). Using the measured TOFs, distances between the reflectors and the vectors of the sensor are calculated, according to Eq. 5.

$$r_{jA} = c \cdot \frac{t_{jA}}{2} \qquad r_{jB} = c \cdot \frac{t_{jB}}{2} \tag{5}$$

  Where t_{jA} and t_{jB} are again the TOFs in vectors defined in Eq. 3; c is the speed of ultrasound in air; and r_{jA} and r_{jB} are distances calculated for the vectors A and B respectively.
- Calculation of reception angles θ_{jA} and θ_{jB} for every vector (tasks U_{jA} and U_{jB}). Considering only vector A, if a reflector P is placed in front of it, two echoes are obtained and the distances r_{j1} and r_{j2} are determined (see Fig. 8). Based on these distances, the reception angle θ_{jA} can be calculated (Eq. 7).

$$r_{j1}^2 = d^2 + r_{j2}^2 - 2 \cdot d \cdot r_{j2} \cdot \cos\left(\frac{\pi}{2} - \theta_{jA}\right) \tag{6}$$

$$\theta_{jA} = \sin^{-1}\left(\frac{d^2 + r_{j2}^2 - r_{j1}^2}{2 \cdot d \cdot r_{j2}}\right) \tag{7}$$

- Reflector classification (task C_j). A first statistical classification of reflectors into two classical types (planes/corners and edges) has been developed, according to [22]. In this way, two statistical functions, f_s (for edges) and f_m (for planes/corners), a constant K, and two standard deviations, σ_s (for edges) and σ_m (for planes/corners), have been defined as follows.

$$f_m = t_{j4}^2 + t_{j1}^2 - t_{j3}^2 - t_{j2}^2 \tag{8}$$

$$f_s = t_{j4}^2 - t_{j3}^2 - t_{j2}^2 + t_{j2} \cdot (t_{j2} + t_{j3} - t_{j4}) \tag{9}$$

$$K = \frac{2 \cdot d \cdot (D + d)}{c^2} \tag{10}$$

$$\sigma_m = \sigma_i \cdot \sqrt{8 \cdot \left(t_{jA}^2 + t_{jB}^2\right)} \tag{11}$$

$$\sigma_s = \sigma_i \cdot \sqrt{4 \cdot (t_{jA} - t_{jB})^2 + t_{jB}^2} \tag{12}$$

  Where σ_i is the standard deviation for an isolated transducer (experimentally determined in 2.5 μs); and the rest of parameters have been defined previously.

Using these definitions, it is possible to classify the detected reflector, following the scheme shown in Fig. 9.

Whenever a reflector is identified as a plane/corner, a second classification process is carried out to distinguish between a plane and a corner. For that purpose, a new classification function f_e and a new constant K_e are defined (Eq. 13).

$$f_e = \theta_{4B} - \theta_{1A}$$

$$K_e = \frac{D + 2 \cdot d}{c \cdot \frac{t_m}{2}} \tag{13}$$

Where θ_{1A} and θ_{4B} are the reception angles of the vectors A and B, for their respective emissions; D and d are the distances determined by the geometry of the sensory structure; c is the propagation speed of ultrasound; and t_m is the averaged TOF for both emissions (Eq. 14).

$$t_m = \frac{t_{11} + t_{12} + t_{13} + t_{14} + t_{41} + t_{42} + t_{43} + t_{44}}{8} \tag{14}$$

According to [22], the function f_e can be used to discriminate between planes and corners, as shown in Eq. 15.

$$\begin{aligned} f_e &= \theta_{4B} - \theta_{1A} = 0 && \text{if} \quad \text{plane} \\ f_e &= \theta_{4B} - \theta_{1A} = K_e && \text{if} \quad \text{corner} \end{aligned} \tag{15}$$

In the same way, Fig. 10 shows a scheme of the classification algorithm used to distinguish between planes and corners.

- Map updating (tasks M_{jA} and M_{jB}). The MHRM method (Heuristic Mapping with Multiple Reception), described in [21], has been used in order to obtain a heuristic map of the environment with 10 × 10 cm cells. The MHRM method is based on the use of templates, which depend on the type of reflector detected. A template is a set of cells, whose certainty values have to be modified due to

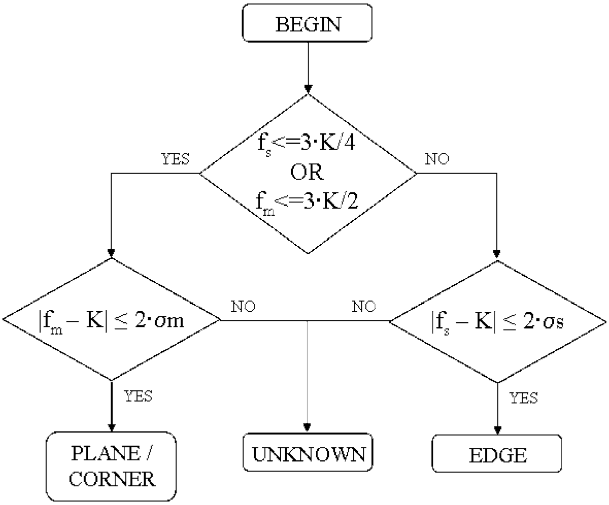
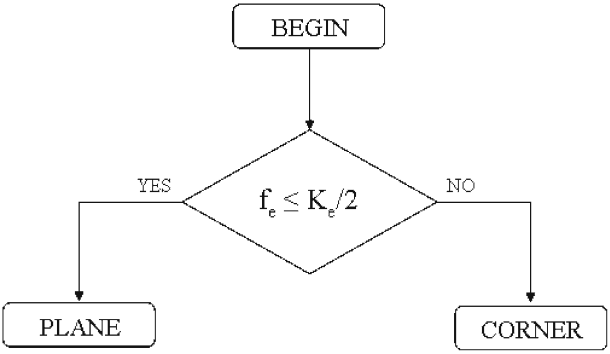


Fig. 9 First step in the reflector classification algorithm: discrimination between planes/corners and edges

Fig. 10 Reflector classification algorithm: second step



the results from the last emission. In these templates, both areas, empty and occupied, are considered and updated at the same time. The certainty value depends on the measured distance, r_{jA} or r_{jB}, and on the reception angle, θ_{jA} or θ_{jB}. The used template function $f(r, \theta)$ is shown in Eq. 16.

$$f(r, \theta) = g(r) \cdot h(\theta) \cdot \mathrm{sign}(r, \theta) \tag{16}$$

Where, the function $\mathrm{sign}(r, \theta)$ implies the difference between an empty and an occupied area, taking into account that only the angular sector at the measured distance r is considered as occupied, whereas the cone between this sector and the sensor is considered as empty.

$$\mathrm{sign}(r, \theta) = \begin{cases} -1 & \text{if} \quad \text{empty} \\ +1 & \text{if} \quad \text{occupied} \end{cases} \tag{17}$$

The functions $g(r)$ and $h(\theta)$ are defined in Eqs. 18 and 19 respectively.

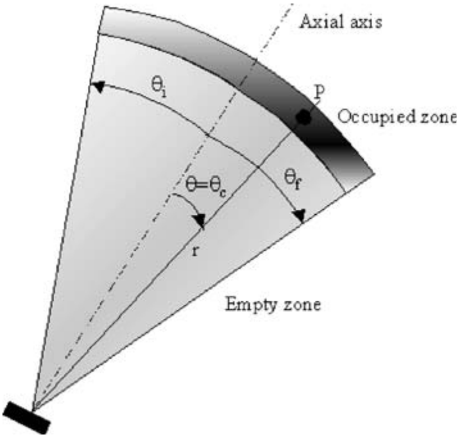
$$g(r) = 1 - \frac{r}{2 \cdot r_{\max}} \qquad\qquad 0 \leq r \leq r_{\max} \tag{18}$$

$$h(\theta) = e^{\left[\frac{(\theta - \theta_c)^2}{2 \cdot \sigma^2}\right]} \qquad\qquad -\frac{\alpha}{2} \leq \theta \leq \frac{\alpha}{2} \tag{19}$$

Where $r_{\max}$ is the maximum measured distance, fixed at 3 m for the near/middle zone; θ_c is the measured reception angle; and σ is the standard deviation of the truncated Gaussian function. The function $g(r)$ provides high values to points near the transducer, whereas low values are given to further distances (1 is given for $r = 0$, and $1/2$ for $r = r_{\max}$). The function $h(\theta)$ assigns a maximum value to the reception angle θ_c, whereas this value decreases whether the angle θ is far away from the reception angle θ_c. The parameter σ determines the slope of this fall.

Furthermore, the MHRM algorithm defines different templates for planes and for edges, so the shape of the template can be adapted better to the detected and classified reflector. Figure 11 shows a generic template defined for edge reflectors. A similar template, but for planes, can be observed in Fig. 12. A matrix composed by 10 × 10 cm cells represents the environment by means of certainty values $V(x, y)$. Whenever the template suitable for a reflector is selected, this

Fig. 11 Generic template for edges defined by the MHRM algorithm



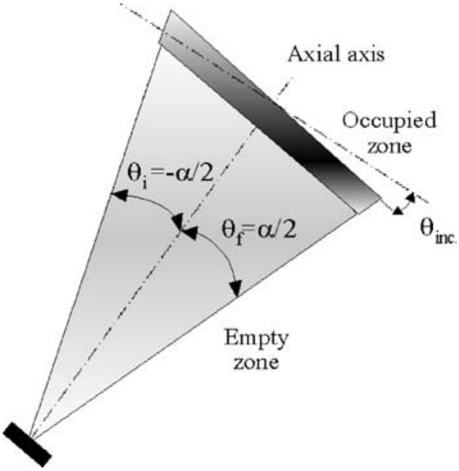
matrix has to be updated with values from that template. This updating process is defined by Eq. 20.

$$V(x, y) = p \cdot f(x, y) + (1 - p) \cdot V(x, y) \qquad -1 \leq V(x, y) \leq +1 \tag{20}$$

Where $f(x, y)$ is the certainty value given by the template; $V(x, y)$ is the certainty value of a cell (x, y); and p fixes a priority between the previous values and the current one.

- Analysis and configuration of the operation mode (tasks E_j and X_j).
  Before next emission, the measured distances r_{jA} and r_{jB} are analysed to change the working zone (near, middle or far), according to the boundaries D_{NM} and D_{MF} mentioned before (see Fig. 7). If necessary, the system is configured and adapted to the new zone, changing the length of the Golay pairs emitted (32, 64 or 128 bits). The new lengths will applied in the next emission process.

Fig. 12 Generic template for planes defined by the MHRM algorithm



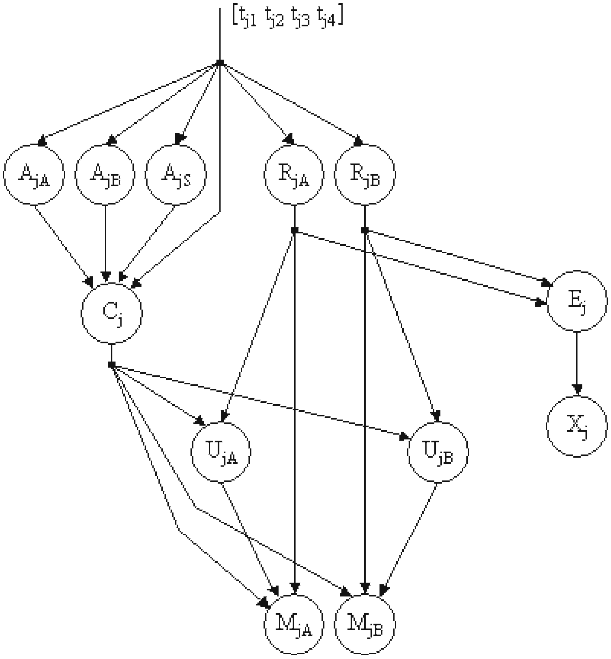


Fig. 13 Dataflow graph for high-level processing in the near and middle zone

In Fig. 13 a dataflow graph is shown that depicts the different tasks for the emission *j*. The representation is useful to analyze dependencies among the different tasks, and their parallelism degree.

3.2 Algorithms for the Far Zone

The detection of further reflectors, with a greater noise influence, implies an increase of the length of the Golay sequences used. The worsening in the probability of the reflector detection, together with constraints on resource utilization (mainly memory capacity to store off-line templates for long distances), imply that the algorithms about reflector classification, or lateral resolution, are not considered. Only an updating of the environment map is carried out. The starting point of the process is again the four TOFs [t_{j1} t_{j2} t_{j3} t_{j4}] from the emission in transducer *j*. In this way, the processing in the far area can be divided into the following steps:

- Determination of distances r_{jA} and r_{jB} between vectors and reflectors (task R_{jA} for vector A and task R_{jB} for vector B).
  This process is same as the distance computing explained in the near and middle zone section.
- Map updating according to the results from the previous step (tasks M_{jA} and M_{jB}).
  The distances obtained before are used to update a heuristic map of the environment in which the detected obstacles are placed. Now, the HAM method (Heuristic Asymmetric Mapping) [8] is used to update it. This method is also based on templates of certainty values, which are generated based only on the measured distance, r_{jA} or r_{jB} (not on the reception angle). There is only one

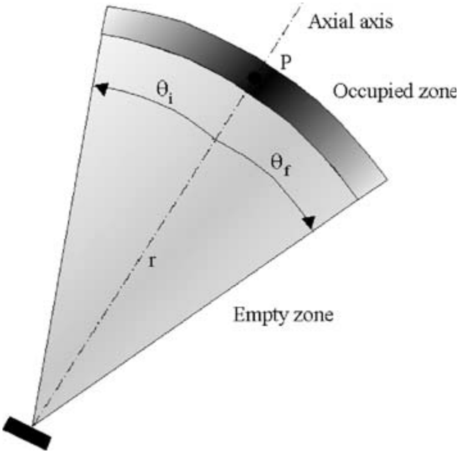


Fig. 14 Generic template defined by the HAM algorithm

type of template (similar to the MHRM edge template), with only a modification in the function $h(\theta)$ since the reception angle θ_c is not available (Eq. 21). Now this function emphasizes the axial axis of the sensor $\theta = 0°$.

$$h(\theta) = e^{\left[\frac{\theta^2}{2\cdot\sigma^2}\right]} \qquad -\frac{\alpha}{2} \leq \theta \leq \frac{\alpha}{2} \tag{21}$$

The last change is about the map resolution, which is decreased to 20 × 20 cm cells. Figure 14 shows a generic template for the HAM method.

- Analysis and configuration of the operation mode (E_j, X_j). Before next emission, the measured distances are analyzed to change the zone (near, middle or

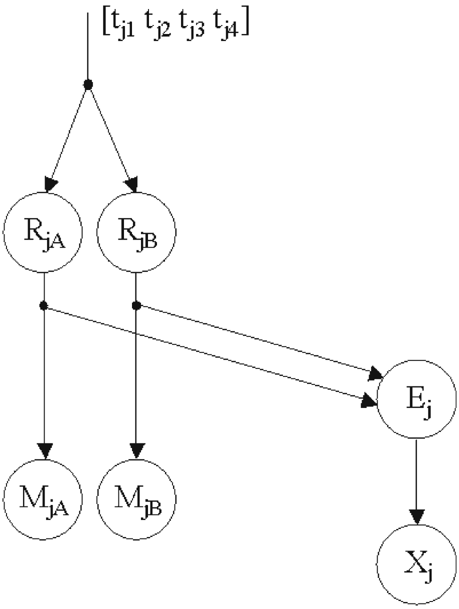


Fig. 15 Dataflow graph for high-level processing in the far zone

far) if necessary. These tasks are similar to those explained in the near and middle zones.

Figure 15 shows the different tasks, keeping their dependencies into consideration, and identifying their execution order.

4 High- and Low-Level Task Integration

In Fig. 16 the scheme of the complete dataflow is shown, developed for the used sensory system. It is divided into low-level and high-level algorithms. In the low-level,

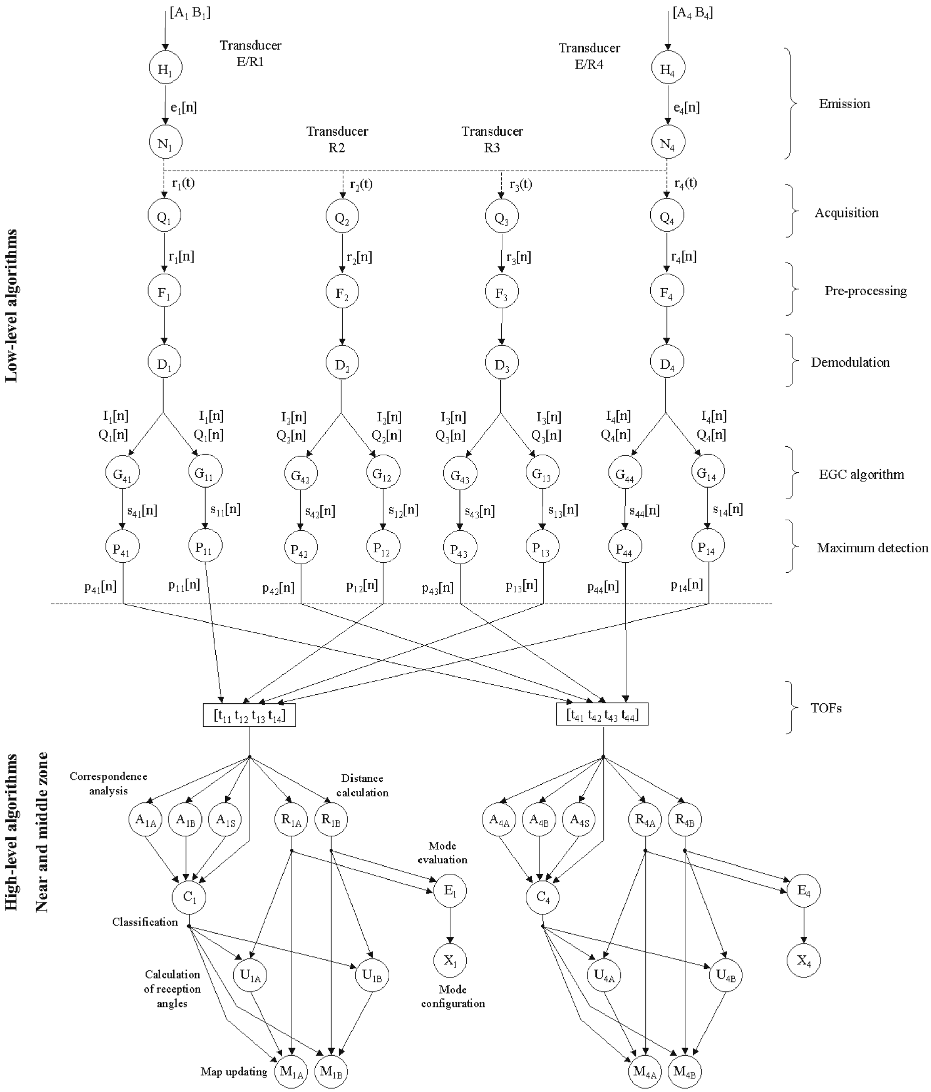


Fig. 16 Complete scheme of the dataflow of the ultrasonic signal processing for the used sensor

the necessary tasks to determine the TOFs are repeated four times (corresponding to the four transducers with simultaneous reception). In the high-level, there are two similar branches that correspond to the similar processing of echoes coming from the emission in the transducers E/R1 and E/R4. The algorithms used in the near and middle zone are represented since they imply more complexity.

A specific hardware platform was used for the implementation of the explained algorithms [43]. Figure 17 shows the general block diagram of the architecture used to implement the algorithms previously described. The architecture can be divided into three different stages:

- Module 1. This is an interface and acquisition system. Every transducer has an interface stage (see Fig. 17) based on the standard system provided by Polaroid for this type of transducers [42]. The signal $r_i(t)$ received by the transducer i ($i = 1, 2, 3, 4$), and amplified by that first stage, is digitized by an acquisition system capable of acquiring signals coming from four different channels (one per transducer) at a frequency $f_S = 400$ kHz.
- Module 2. This is the first processing stage, based on a Texas Instruments C6701 DSP with a clock frequency of 166 MHz. In this way, it can receive the samples acquired by the four transducers to process them, either individually, or in cooperation with the third block of the system.
- Module 3. This is based on a FPGA device from the Xilinx Virtex family, with a main clock frequency of 50 MHz and connected to the previous DSP by means of two communication ports.

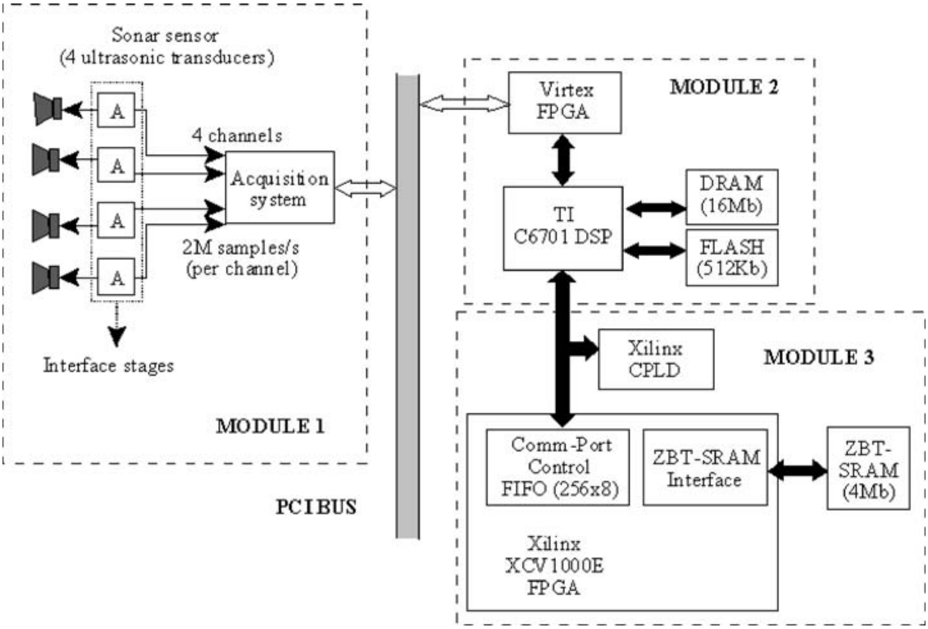


Fig. 17 Block diagram of the used hardware architecture

The implementation of the different tasks is distributed between the FPGA device and the DSP module, depending on their computational requirements. Low-level processing for TOF determination consists of demodulation and correlation. Since the encoding uses binary sequences, these processes are reduced to only additions and subtractions. In that way, the FPGA device is suitable for a real-time and high-performance implementation of the low-level processing [23].

On the other hand, high-level processing requires more complex computing, often based on multiplications and statistical and/or trigonometric functions. In this case, the DSP is a better choice for a real-time implementation. However, the DSP also has some performance constraints, referred to the map updating tasks. Since the real-time computing of the different certainty values to apply to the environment map after every emission is not feasible, the proposed solution generates the updating templates off-line, and store them in a DRAM bank. In order to fit in the available memory capacity, the generated templates have been discretized to defined standard values. In the near/middle zone, templates are generated at every 10 cm and for reception angles every 7.5°, from $-15°$ to $+15°$; whereas in the far zone, they are generated every 20 cm with null reception angles (0°).

Regarding the power consumption, it is important to remark that the FPGA and the DSP devices are the most significant, apart from the acquisition stage. Assuming the worst case, the acquisition module requires 9.1 W, whereas the DSP and the FPGA consume 6 and 4 W respectively.

5 Computing Times and Temporal Distribution

The global computing time for the different algorithms developed for the sonar system depends on the working zone, near/middle or far. Table 1 shows the times obtained for the different tasks on the different computing platform described before. The tasks have been developed either in the FPGA or in the DSP, based on their data and computational requirements. Low-level processing (TOF determination) can be only implemented on FPGA-based platforms [44], whereas high-level processing (classification reflector and mapping) is more suitable for a DSP device, due to the mathematical computing required (trigonometry and statistics).

Considering the global task diagram (see Fig. 16) and Table 1, the global computing time can be obtained for the near and middle zone $t_{\text{total}}^{\text{NM}}$ in Eq. 22 and for the far zone t_{total}^{F} in Eq. 23. Note that times described in Table 1 refer to only one emission

Table 1 Computing times for tasks carried out in the ultrasonic signal processing

	Computing times	Device
TOF determination for a transducer *i*	323.40 μs	FPGA
Correspondence analysis for an emission *j*	2.42 μs	FPGA
Reflector classification for an emission *j*	238.41 μs	DSP
Distance and angle calculation for a vector (near/middle–far)	33.57–17.66 μs	DSP
Map updating for a vector (near/middle–far)	8.24–15.19 ms	DSP
Mode evaluation	10.19 μs	DSP
Mode configuration	0.74 μs	DSP

(two simultaneous emissions in E/R1 and E/R4 exist), to only one transducer (in the case of the low level), and to a vector (every emitting process provides data in both vectors).

$$t_{\text{total}}^{\text{NM}} = 4 \cdot 323.4\ \mu\text{s} + 2 \cdot 2.42\ \mu\text{s} + 2 \cdot 238.41\ \mu\text{s} + 4 \cdot 33.57\ \mu\text{s} + \\ + 4 \cdot 8.24\ \text{ms} + 10.19\ \mu\text{s} + 0.74\ \mu\text{s} = 34.87\ \text{ms} \tag{22}$$

$$t_{\text{total}}^{\text{F}} = 4 \cdot 323.4\ \mu\text{s} + 2 \cdot 2.42\ \mu\text{s} + 2 \cdot 238.41\ \mu\text{s} + 4 \cdot 17.66\ \mu\text{s} \\ + 4 \cdot 15.19\ \text{ms} + 10.19\ \mu\text{s} + 0.74\ \mu\text{s} = 62.62\ \text{ms} \tag{23}$$

Regarding the resource consumption of the Virtex FPGA, the number of slices used is 3,736 (30.4% in a Xilinx Virtex XCV1000E), assuming the worst case: use of 128-bit Golay complementary pairs. The design in the FPGA is capable of operating at a frequency of 50 MHz.

Due to the characteristics of ultrasonic transducers, successive emissions are usually delayed by an interval, that is typically between 80 and 100 ms. These global times allow to obtain a system with an emission frequency around 16 Hz (16 emission processed per second), high enough for the previously mentioned intervals.

6 Results

Some results from the classification analysis carried out are shown in Tables 2 and 3 for an edge and a plane, respectively, at different positions. The accuracy of the low-level processing has already been detailed in [23], achieving figures below 1 mm. Note that, in the cases marked with an asterisk in these Tables, the reflector is out of the range of one vector, so it provides high percentages of unknown reflectors.

Table 2 Classification results for an edge reflector after 500 emissions

	Emitter (%)	Unknown (%)	Edge (%)	Plane (%)	Corner (%)
Position (distance r, angle θ)					
($r = 100$ cm, $\theta = 0°$)	E/R1	10	90	0	0
($r = 100$ cm, $\theta = 0°$)	E/R4	11	89	0	0
*($r = 100$ cm, $\theta = 5°$)	E/R1	100	0	0	0
($r = 100$ cm, $\theta = 5°$)	E/R4	0	99	1	0
($r = 100$ cm, $\theta = -5°$)	E/R1	1	99	0	0
*($r = 100$ cm, $\theta = -5°$)	E/R4	100	0	0	0
Global percentages at $r = 100$ cm	Both	37	62.8	0.2	0
($r = 200$ cm, $\theta = 0°$)	E/R1	20	70	10	0
($r = 200$ cm, $\theta = 0°$)	E/R4	11	89	0	0
($r = 200$ cm, $\theta = 5°$)	E/R1	43	50	7	0
($r = 200$ cm, $\theta = 5°$)	E/R4	21	64	15	0
($r = 200$ cm, $\theta = -5°$)	E/R1	10	83	7	0
($r = 200$ cm, $\theta = -5°$)	E/R4	35	65	0	0
Global percentages at $r = 200$ cm	Both	23.3	70.2	6.5	0

Table 3 Classification results for a plane reflector after 500 emissions

	Emitter (%)	Unknown (%)	Edge (%)	Plane (%)	Corner (%)
Position (distance r, inclination θ)					
($r = 150$ cm, $\theta = 0°$)	E/R1	4	2	94	0
($r = 150$ cm, $\theta = 0°$)	E/R4	1	1	98	0
($r = 150$ cm, $\theta = 5°$)	E/R1	26	9	65	0
*($r = 150$ cm, $\theta = 5°$)	E/R4	100	0	0	0
*($r = 150$ cm, $\theta = -5°$)	E/R1	100	0	0	0
($r = 150$ cm, $\theta = -5°$)	E/R4	11	1	88	0
Global percentages at $r = 150$ cm	Both	40.3	2.2	57.5	0
($r = 250$ cm, $\theta = 0°$)	E/R1	13	17	70	0
($r = 250$ cm, $\theta = 0°$)	E/R4	4	20	76	0
($r = 250$ cm, $\theta = 5°$)	E/R1	16	22	62	0
*($r = 250$ cm, $\theta = 5°$)	E/R4	48	19	33	0
*($r = 250$ cm, $\theta = -5°$)	E/R1	47	12	41	0
($r = 250$ cm, $\theta = -5°$)	E/R4	39	19	42	0
Global percentages at $r = 250$ cm	Both	27.8	18.2	54	0

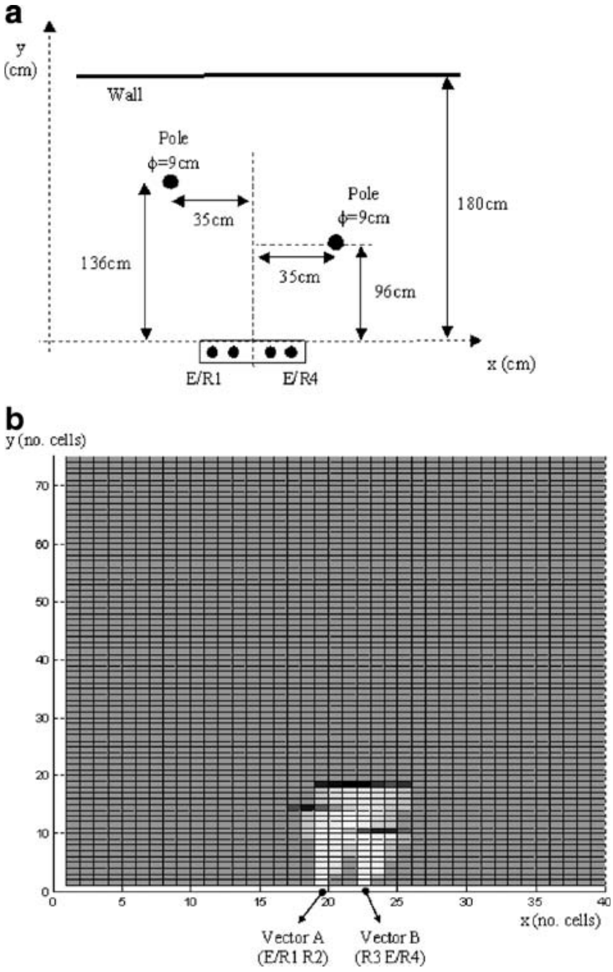


Fig. 18 Test no. 1: **a** Geometrical distribution; **b** results obtained after only one emission

Some practical tests have been carried out to check the feasibility in mapping applications. For test no. 1, in the scenario shown in Fig. 18a, the environment is divided into a Cartesian grid of 10 × 10 cm cells, in such a way that the vector A (transducers E/R1 and R2) is located at the coordinate (19,1) and the vector B (transducers R3 and E/R4) at the coordinate (22,1)—both on the X-axis. After only one emission, the resulting map can be observed in Fig. 18b. In this representation, the certainty values (CVs) have been scaled to the interval [0, 1]—see Eq. 24. In this case, between the plane and the sensor, there are two identical poles (with a diameter of $\phi = 9$ cm) separated from the axial axis, one towards the vector A and the other one towards the vector B. The three reflectors are correctly detected and represented in the environment map. It is important to remark that this map is obtained after only one emission process (around 62 ms, computational time for the proposed implementation). These results provide more information from the environment in

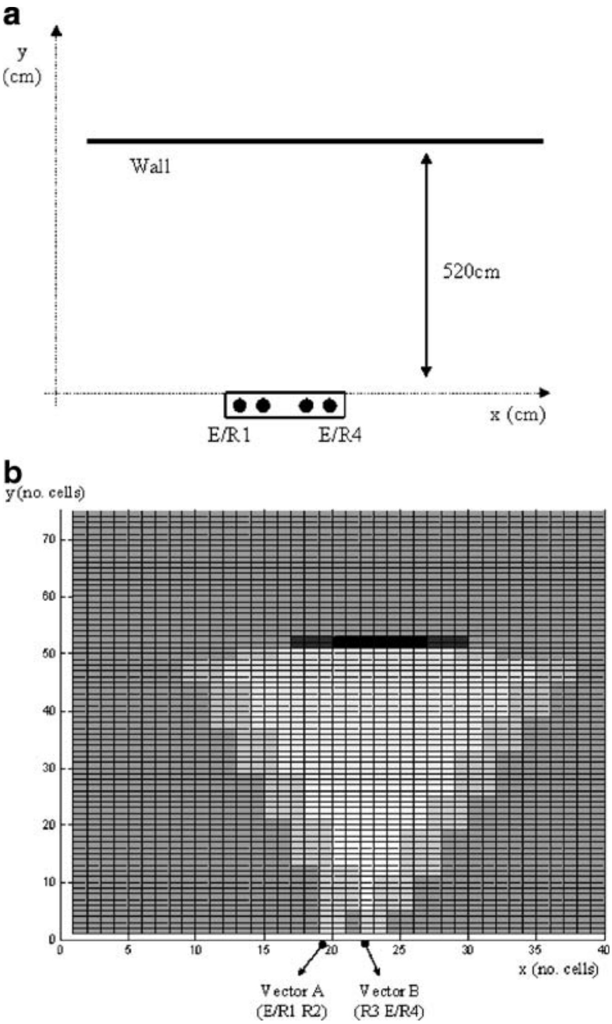


Fig. 19 Test no. 2: **a** Geometrical distribution; **b** results obtained after only one emission

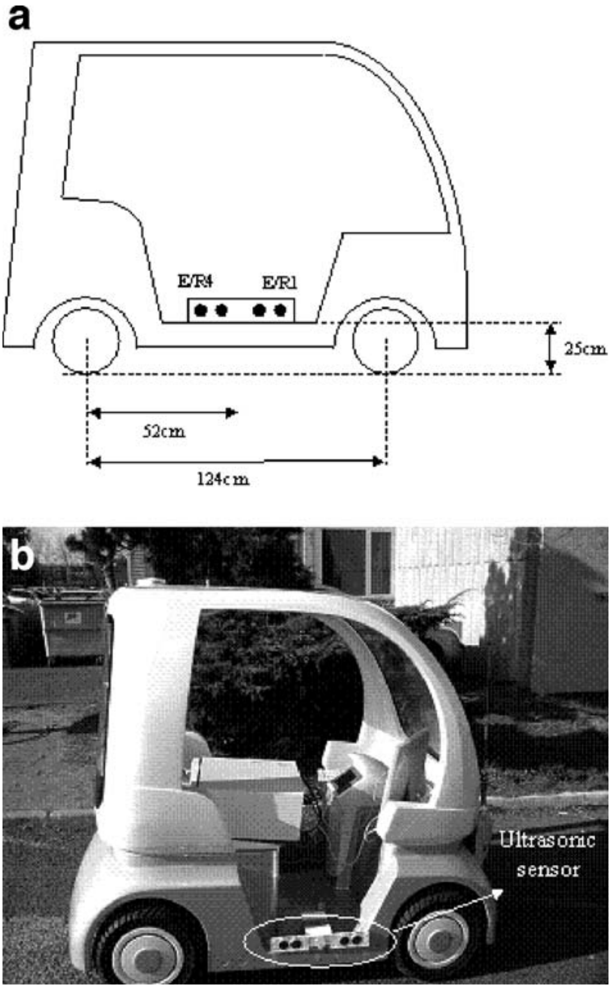


Fig. 20 Scheme of the placement of the ultrasonic sensor in the CyCab

only one emission, compared to simple sonar systems, where reflectors are often given on the axial axis of an isolated transducer [8, 11, 13].

$$\begin{array}{ll} 0 \leq \mathrm{CV} < 0.5 & \text{for empty zone (clear color)} \\ \mathrm{CV} = 0.5 & \text{for unknown zone (grey color)} \\ 0.5 < \mathrm{CV} \leq 1.0 & \text{for occupied zone (dark color)} \end{array} \tag{24}$$

In Fig. 19 test no. 2 is shown, where a plane has been placed in the far zone. It can be observed its correct detection, and how the dimensions of updated cells are the corresponding 20 × 20 cm.

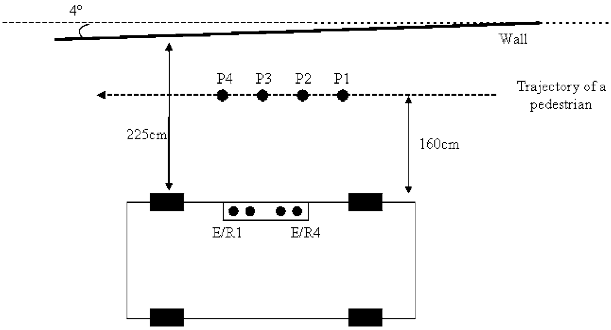


Fig. 21 Positions analysed in front of the sensory structure for a pedestrian

6.1 Installation on a CyCab Vehicle

With the aim of checking the sensory structure on a vehicle, some experimental tests have been carried out on a CyCab vehicle by Robosoft [45]. Figure 20 shows the dimensions of the vehicle, and the position of the sensor in its structure. Tests have been carried out on a sunny day, with a temperature of approximately 17°C, windless, and without any other remarkable meteorological interference.

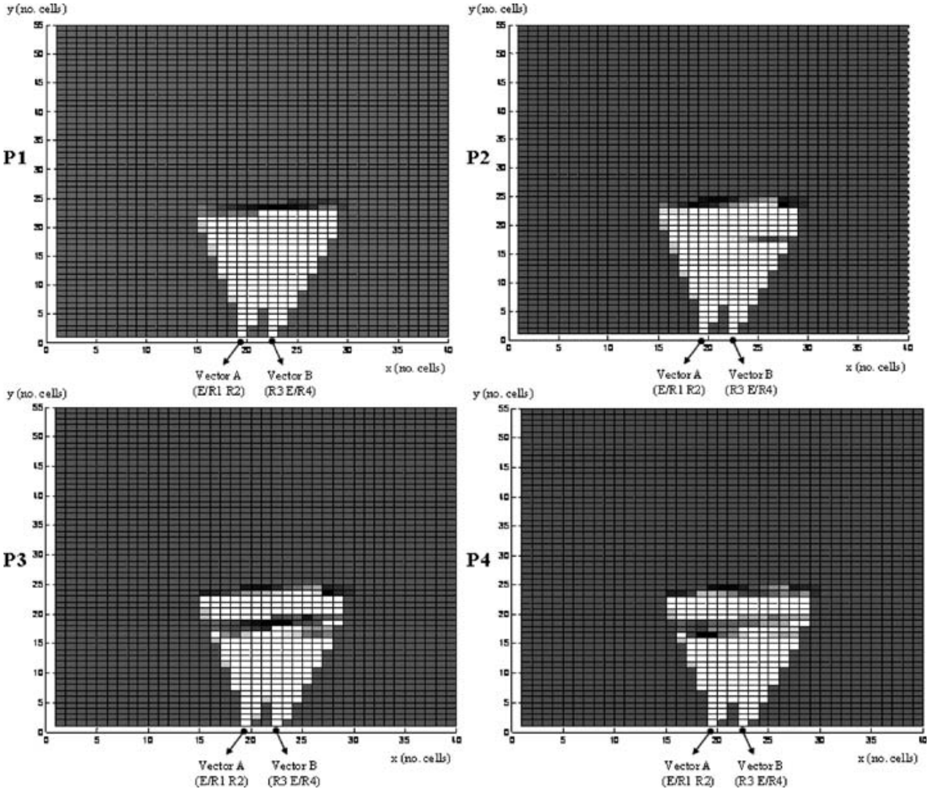


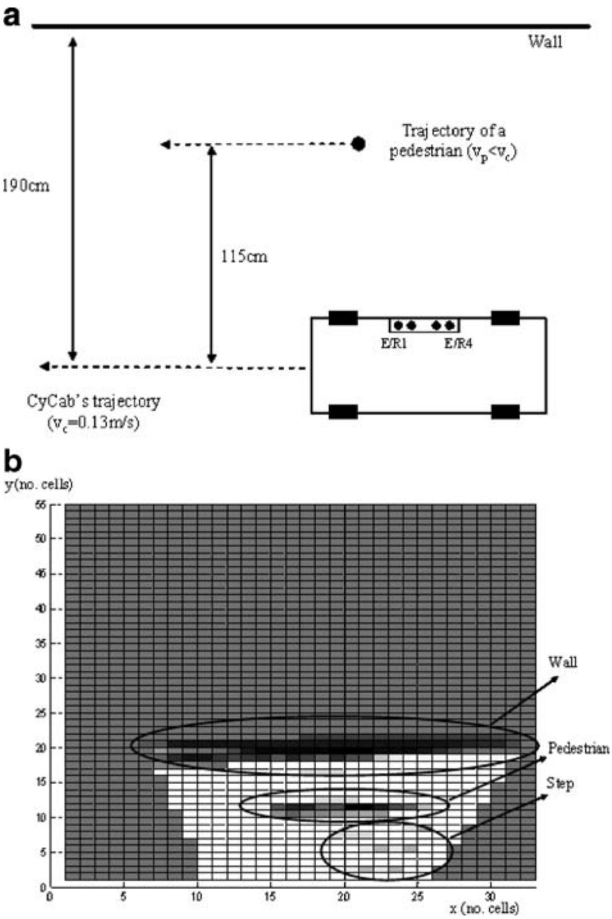
Fig. 22 Experimental results for the positions considered in Fig. 21

In the first experiment the vehicle was kept static and a pedestrian was made to pass in front of the sensor, so four emission processes were analysed in positions P1, P2, P3 and P4 (see Fig. 21). Figure 22 shows the results for each position, where it can be observed how the back wall and the pedestrian were detected, as well as how fast a map can be updated from an emission to the next one.

Figure 23a shows an example where the vehicle moves with a speed of 13 cm/s, whereas the pedestrian walks more slowly. The pedestrian and the CyCab move in the same direction, so he is reached and overtaken by the vehicle. The results obtained in Fig. 23b depict the pedestrian's position, and how this changes along time. The step existing between the pedestrian and the vehicle can also be identified. This step provides not only the direct reflection but also the corresponding multiple reflections between step and vehicle.

Figure 24 shows a third situation, where the pedestrian follows an elliptical trajectory towards the vehicle. Again, the back wall and the pedestrian are correctly detected, so are the step and its multiple reflections. Furthermore, a wrong echo is mapped behind the wall, caused by a multiple reflection between the wall, the pedestrian and the vehicle.

Fig. 23 Test no. 3: **a** Geometrical distribution; **b** experimental results



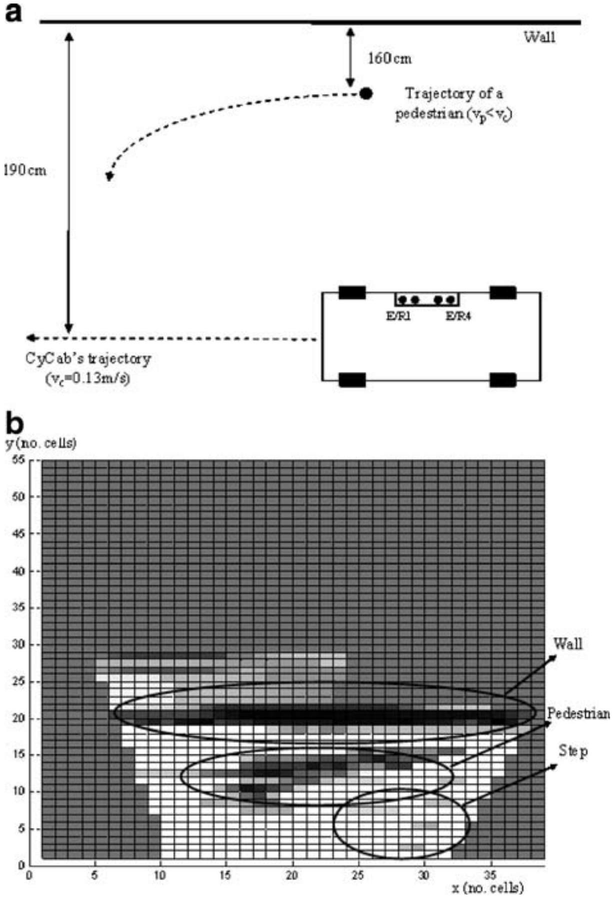


Fig. 24 Test no. 4: **a** Geometrical distribution; **b** experimental results

7 Conclusions

A real-time implementation of the processing algorithms for an advanced sonar module has been presented. The implementation is based on a DSP–FPGA architecture. An accurate low-level processing has been developed, consisting of the encoding by Golay complementary sequences. This provides the possibility of implementing ultrasonic arrays with simultaneous reception in all the transducers, and simultaneous emission in at least two of them without crosstalk interferences. These features allow a complex high-level processing to be developed, where a heuristic map of the environment is built. The way this map is updated and obtained is adaptable to different circumstances and necessities. Algorithms and methods are modified in real-time, trying to obtain the best performances from the measurements and to achieve an optimal use of the resources in the implementation. The system proposed here is capable of computing all the algorithms from both processing levels in real-time, obtaining accurate results and updating an environment map between consecutive emissions. Finally, the sonar system has been installed on a CyCab vehicle, where its feasibility has been verified in some tests. The proposed sonar module and the developed computing platform provides a suitable solution for robotics applications, in mapping or obstacle detection.

Acknowledgements This work has been possible thanks to the Ministerio de Educación y Ciencia of Spain (RESELAI project, ref. TIN2006-14896-C02-01), to the Comunidad de Madrid and the University of Alcalá (INCUBUS project, ref. CCG06-UAH/DPI-0725), to the Ministerio de Trabajo y Asuntos Sociales of Spain (DEDALO project, ref. 81/06), and to the Regional Council of Auvergne in France (Contrat de Plan Etat/Région).

References

1. Borenstein, J., Koren, Y.: Obstacle avoidance with ultrasonics sensors. IEEE J. Robot. Autom. **4**(2), 213–218 (1988)
2. Everett, H.R.: Sensors for Mobile Robot: Theory and Application. A K Peters Ltd., Wellesley, M. A. (1995)
3. Song, K.T., Chen, C.H., Huang, C.H.C.: Design and experimental study of an ultrasonic sensor system for lateral collision avoidance at low speeds. In: Proc. 2004 IEEE Intelligent Vehicles Symposium (IV'04), pp. 647–652. Parma, Italy (2004)
4. Del Castillo, G., Skaar, S., Cardenas, A., Fehr, L.: A sonar approach to obstacle detection for a vision-based autonomous wheelchair. Robot. Auton. Syst. **54**(12), 967–981 (2006)
5. Jorg, K., Berg, M.: Mobile robot sonar sensing with pseudo-random codes. In: Proc. 1998 IEEE Conference on Robotics and Automation (ICRA'98), pp. 2807–2812. Leuven, Belgium (1998)
6. Kuc, R., Barshan, B.: Bat-like sonar for guiding mobile robots. IEEE Control Syst. **23**(2), 4–12 (1992)
7. Jiménez, A.R., Seco, F.: Precise localisation of archaeological findings with a new ultrasonic 3D positioning sensor. Sens. Actuators A, **123–4**, 224–233 (2005)
8. Song, K.T., Chen, C.C.: Application of heuristic asymmetric mapping for mobile robot navigation using ultrasonic sensors. J. Intell. Robot. Syst. **17**, 243–264 (1996)
9. Barshan, B., Ayrulu, B.: Performance comparison of four time-of-flight estimation methods for sonar signals. Electron. Lett. **34**(16), 1616–1617 (1998)
10. Peremans, H., Audenaert, K., Van Campenhout, J.: A high resolution sensor based on tri-aural perception. IEEE Trans. Robot. Autom. **9**(1), 36–48 (1993)
11. Borenstein, J., Koren, Y.: Histogramic in-motion mapping for mobile robot obstacle avoidance. IEEE Trans. Robot. Autom. **7**(4), 535–539 (1991)
12. Chong, K.S., Kleeman, L.: Mobile robot map building for an advanced sonar array and accurate odometry. Int. J. Rob. Res. **18**(1), 20–36 (1999)
13. Elfes, A.: Occupancy grids: a stochastic spatial representation for active robot perception. In: Proc. 6th IEEE Conf. on Uncertainty in AI. Cambridge, USA (1990)
14. Kleeman, L.: Advanced sonar with velocity compensation. Int. J. Rob. Res. **23**(2), 111–126 (2004)
15. Kuc, R.: Pseudo-amplitude scan sonar maps. IEEE Trans. Robot. Autom. **17**(5), 767–770 (2001)
16. Wijk, O., Christensen, H.: Triangulation based fusion of sonar data with application in robot pose tracking. IEEE Trans. Robot. Autom. **16**(6), 740–752 (2000)
17. Choset, H., Nagatini, K., Lazar, N.A.: The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy. IEEE Trans. Robot. Autom. **19**(3), 513–522 (2003)
18. Benet, G., Martínez, M., Blanes, F., Pérez, P., Simó, J.E.: Differentiating walls from corners using the amplitude of ultrasonic echoes. Robot. Auton. Syst. **50**, 13–25 (2005)
19. Moita, F., Lopes, A.C., Nunes, U.: A fast firing binaural system for ultrasonic pattern recognition. J. Intell. Robot. Syst. **50**(2), 141–162 (2007)
20. Peremans, H.: A maximum likelihood algorithm for solving the correspondence problem in tri-aural perception. In: Proc. 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'94), pp. 485–492. Las Vegas, USA (1994)
21. Ureña, J., Mazo, M., García, J.J., Bueno, E., Hernández, A., Hernanz, D.: Low-cost improvement of an ultrasonic sensor and its characterization for map building. In: Proc. IFAC Workshop on Intelligent Components for Vehicles (ICV98), pp. 333–338. Seville, Spain (1998)
22. Ureña, J., Mazo, M., García, J.J., Hernández, A., Bueno, E.: Classification of reflectors with an ultrasonic sensor for mobile robot applications. Robot. Auton. Syst. **29**, 269–279 (1999)

23. Hernández, A., Ureña, J., García, J.J., Mazo, M., Herranz, D., Dérutin, J.P., Sérot, J.: Ultrasonic ranging sensor using simultaneous emissions from different transducers. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **51**(12), 1660–1670 (2004)
24. Álvarez, F.J., Ureña, J., Mazo, M., Hernández, A., García, J.J., De Marziani, C.: High reliability outdoor sonar prototype based on efficent signal coding. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **53**, 1863–1870 (2006)
25. Barshan, B., Kuc, R.: Differentiating sonar reflections from corners and planes by employing an intelligent sensor. IEEE Trans. Pattern Anal. Mach. Intell. **12**(6), 560–568 (1990)
26. Kleeman, L., Kuc, R.: Mobile robot sonar for target localization and classification. Int. J. Rob. Res. **14**(4), 295–318 (1995)
27. Gao, W., Hinders, M.K.: Mobile robot sonar interpretation algorithm for distinguishing trees from poles. Robot. Auton. Syst. **53**(2), 89–98 (2005)
28. Lee, C., Yan, L., Lee, S.: Estimation of inclination angle of a planar reflector using sonar sensor. IEEE Sens. J **7**(7/8), 1052 (2007)
29. Kleeman, L.: Advanced sonar and odometry error modeling for simultaneous localisation and map building. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 699–704. Las Vegas, USA (2003)
30. Karaman, O., Temeltas, H.: Comparison of different grid based techniques for real-time map building. In: Proc. 2004 IEEE International Conference on Industrial Technology (ICIT'04), vol. 2 (2004)
31. Noykov, S., Roumenin, C.: Calibration and interface of a Polaroid ultrasonic sensor for mobile robots. Sens. Actuators, A, Phys. **135**(1), 169–178 (2007)
32. Leonard, J.J., DurrantWhyte, H.F.: Simultaneous map building and localization for an autonomous mobile robot. In: IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS'91), vol. 3, pp. 1442–1447. Osaka, Japan (1991)
33. Newman, P., Cole, D., Ho, K.: Outdoor SLAM using visual appearance and laser ranging. In: IEEE International Conference on Robotics and Automation (ICRA'06). Orlando, USA (2006)
34. Davison, A.J., David, W.M.: Simultaneous localization and map-building using active vision. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 865–880 (2002)
35. Audenaert, K., Peremans, H., Kawahara, Y., Van Campenhout, J.: Accurate ranging of multiple objects using ultrasonic sensors. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'05), pp. 1733–1738. Nice, France (1992)
36. Graham, P., Nelson, B.: Frequency-domain sonar processing in FPGAs and DSPs. In: Proc. IEEE Symposium on FPGAs for Custom Computing Machines (FCCM'98), pp. 306–307. Napa Valley CA, USA (1998)
37. Nelson, B.E.: Configurable computing and sonar processing—architectures and implementations. In: Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 56–60. Pacific Grove CA, USA (2001)
38. Clarke, C., Qiang, L., Peremans, H., Hernández, A.: FPGA implementation of a neuromimetic cochlea for a bionic bat head. In: Proc. 14th International Conference on Field-Programmable Logic and its applications (FPL'04), pp. 1073–1075. Antwerp, Belgium (2004)
39. Lee, B.B., Furgason, E.S.: High speed digital Golay code flaw detection system. Ultrasonics **21**(4), 153–161 (1983)
40. Budisin, S.Z.: Efficient pulse compressor for Golay complementary sequences. Electron. Lett. **27**(3), 219–220 (1991)
41. Popovic, B.M.: Efficient Golay correlator. IEE Electron. Lett. **35**(17), 1427–1428 (1999)
42. Polaroid Corporation: 600 series. Instrument grade electrostatic transducers. Technical specification (1999)
43. Sundance Multiprocessor Technology Ltd.: SMT320v4, SMT358, SMT335-375 User guide. User manual (2003)
44. Hernández, A., Urena, J., Mazo, M., Jiménez, A., Garcá, J.J., Álvarez, F.J., De Marziani, C., Ochoa, A.M.: A comparison of computing architectures for ultrasonic signal processing. In: Proc. IEEE Int. Symposium on Intelligent Signal Processing (WISP'05). Faro, Portugal (2005)
45. Robosoft: ROBOSOFT's intelligent electric vehicles. Product specification (2003)