Audio Source Localization, using a Network of Embedded Devices

Laurențiu FRANGU, Marius MĂZĂREL, Claudiu CHICULIȚĂ
Dunărea de Jos University of Galați
str.Domnească 47, RO-800008 Galați
Laurentiu.Frangu@ugal.ro

***Abstract*—In this paper, a problem of audio source localization is solved, using a network of embedded devices. The intensive computing procedures (such as the crosscorrelation functions) are performed by the embedded devices, which have enough speed and memory for this task. A central computer computes the position in a fast procedure, using the data transmitted by the network nodes, and plays the role of operator interface. The paper also contains the description of the embedded devices, which are designed and manufactured by the authors. They prove to be suited for this kind of application, as they perform fast computation and require low power and small space for installing.**

***Index Terms*—audio source, direction of arrival (DoA), embedded devices, position measurement, sensor network**

I. Introduction

Audio source localization is one of the problems to be solved in surveillance systems. The paper presents an audio source localization application and the embedded devices required for this application. Its objective is to describe the method used for localization and to evaluate the performance of the network in such an application.

The localization problem was addressed by many works (see [1] for DoA). They describe the method for measuring the direction of arrival of sound (DoA), for devices using a sensor array. The delay between the sound recorded by more receivers (microphones) is usually determined by the crosscorrelation function. In this work, the mentioned method is used for measuring a 2-coordinate position of the audio source. The signals are received by more sensor arrays, each of them being associated with an embedded signal processing device.

The devices are organized in a sensor network. They process audio and video data and transmit the results to a central computer. Within this application, the main processing result on each node of the network is the direction of arrival of the sound, whereas the central computer computes the coordinates of the audio source. A typical scene when using the network to the source localization is presented in fig. 1, where the parameters describe the position of each node.

The devices are designed by the authors and are intended for data capture and signal processing, mainly in the field of video and audio signals. The network of embedded devices was previously described in [2], [3]. Other possible applications of such devices were described in the fields of video detection ([4]) and threat evaluation ([5]).

The paper is organized as follows: section II introduces the embedded device, section III describes the method used for localization, and section IV is reserved for experimental results and conclusions.

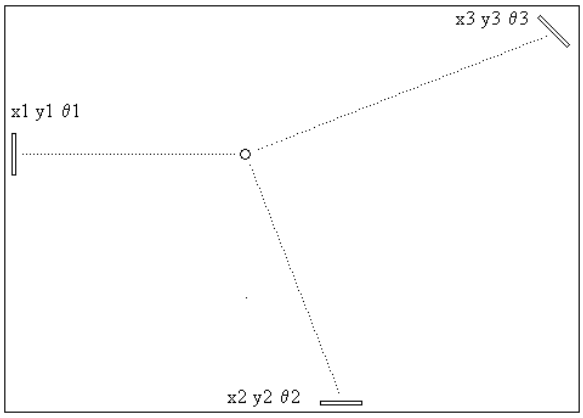

Figure 1. A typical scene when using the network of embedded devices.

II. The Embedded Devices

The application uses a network of embedded devices for data capture and signal processing and a central computer as system-human interface. The network is composed of several embedded devices (the *nodes*), that communicate either by wired or wireless Ethernet. The main components of a node are:

- the video processing subsystem
- the audio processing subsystem
- the reasoning and decision board
- the wireless module
- the communication and power board (motherboard).

The block diagram of the node is presented in fig. 2 (as introduced in [3]). The hardware resources exploited in this work are situated on the audio board, which uses two Blackfin processors. The block diagram of the audio subsystem is presented in fig. 3 (as introduced in [3]).

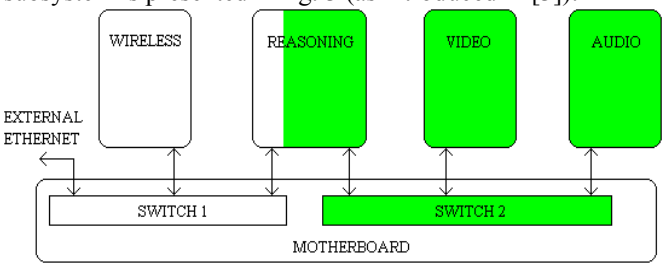

Figure 2. The block diagram of the node.

The audio processing subsystem is intended to capture sound through the microphones, to extract low level features from the audio signals and to send this information to the

Digital Object Identifier 10.4316/AECE.2010.02023

reasoning and decision unit. The main characteristics are:

- 8 microphones, spaced at 30mm
- variable gain, in the range of 10 – 2700
- cut-off frequency of the 2nd order lowpass filter: 22 kHz
- sampling frequency: up to 375 kHz for all inputs (usually at 44 kHz)
- ADC resolution: 12 bits
- 2 signal processing modules, based on Blackfin DSPs
- own noise: < 0.1%
- external connections: 8 microphone inputs, Ethernet, JTAG, 2 UART
- dimensions of the board: 100mm x 70mm, 100g
- power consumption: approx. 2.5W.

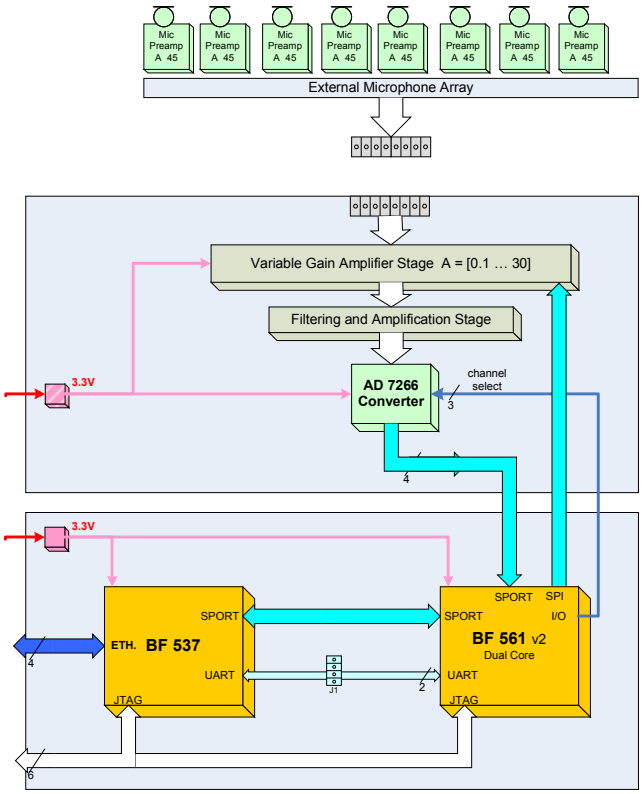

Figure 3. The block diagram of the audio subsystem.

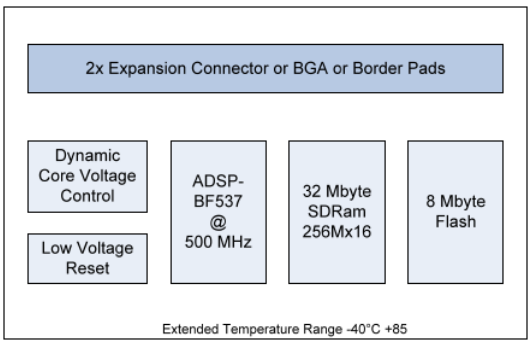

Figure 4. Structure of the BF 537 core module.

The two processing modules (manufactured by Bluetechnix, Austria) are based on the Blackfin processors, BF561 and BF537. The structure of the CM-BF537 module is presented in fig. 4. The features of the module:

- Analog Devices Blackfin processor BF537, 500MHz
- 32 MB SDRAM, 8 MB of Addressable Flash, SDRAM clock up to 133MHz
- small size: 36.5 x 31.5 mm.
- The board is endowed with basic input/output functions:
- audio data acquisition and data transfer to BF561
- data transfer between the two processors (UART, at 7Mbps or SPORT, at 30 Mbps)
- Ethernet communication through BF537 (maximum flow rate 1.6 MBytes/s).

The other core module is based on the BF561 processor. The main difference between the processors is that BF561 contains two processing cores and does not include an Ethernet interface. The SDRAM memory on the module is 64 MB. The rest of the properties are similar to CM-BF537.

The two cores of BF561 are a very useful feature: the data acquisition on 8 channels and data processing can be performed simultaneously, without dead times, which is essential in sound processing. For data processing speed reference, a crosscorrelation function for 1024 samples (that is 25 ms of recorded sound) takes 5.3 ms on BF561.

The audio processing board and the assembled node are presented in figures 5 and 6, respectively.

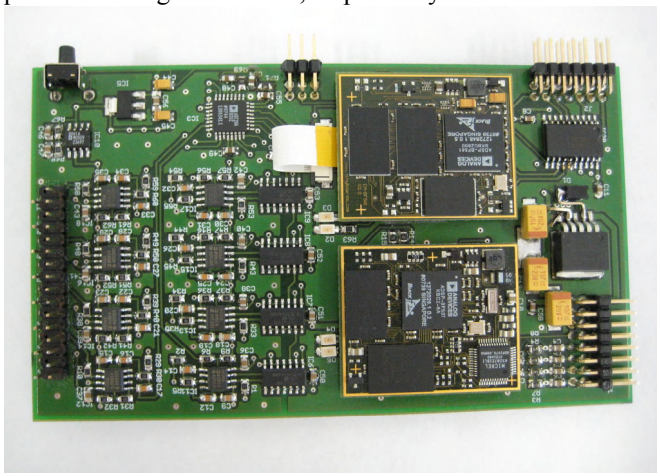

Figure 5. The audio processing board.

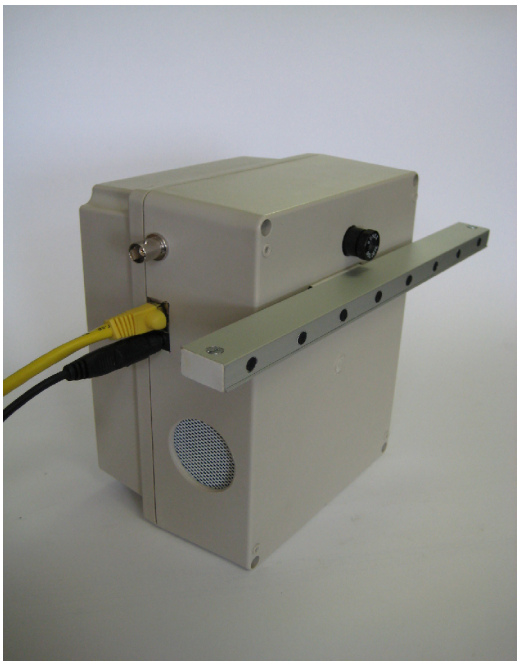

Figure 6. The node, with apparent camera and microphone array.

III. Audio Source Localization

The principle of the source localization is based on the computation of the DoA by each node. The position of the source is then determined in the central computer using these values and the geometrical relations between them. For making the ideas clear, some simplifying assumptions are made: there is only one source in the area, its level is permanently above the noise and the source and the microphone arrays are considered to stay in the same horizontal plane.

A. Measuring the DoA

The direction of arrival of the sound to the node (DoA) is defined with respect to the normal of the microphone array, as illustrated in fig. 7 (audio source figured as a cross and microphones figured as circles). In the following, the variable α_k denotes the DoA computed at node k. Measuring DoA requires to determine the delay between the time of arrival of the sound to the microphones. For simplicity, only two microphones are considered, although using more microphones reduces the error produced by the noise and the multipath propagation. The distance d between the extreme microphones is 210 mm, the sound speed c is assumed to be known, so the direction is determined according to relation (1):

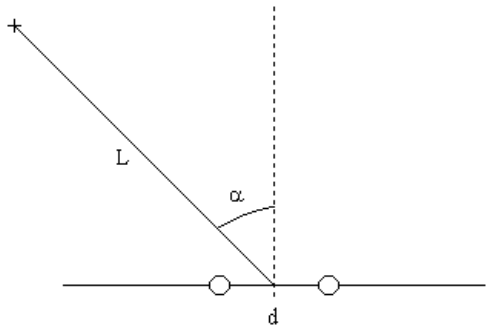

Figure 7. Schematic of the sound arrival (DoA with respect to the normal of the microphone array).

$$\alpha = \arcsin\frac{c\cdot\Delta t}{d} \tag{1}$$

where Δt is the delay between the signals recorded by the two microphones. These signals are assumed to contain the delayed version of the source signal and a noise component, as in the relations (2):

$$r_1(k) = a_1\cdot s(k-D_1)+z_1(k)$$
$$r_2(k) = a_2\cdot s(k-D_2)+z_2(k) \tag{2}$$

where r is the recorded signal, s is the original signal, D is the delay to the reception and z is a white noise. All signals are sampled and the time k is an integer, expressed in sampling periods. The two sequences of the noise are assumed to be uncorrelated with the signal and with each other (see [1]). Consequently, the last three components of the crosscorrelation function of the recorded signals, as expressed in relation (3), will produce a negligeable contribution (they approach zero, as the length of the recorded signals increases).

$$R_{r1r2}(\tau) = a_1a_2R_{ss}(\tau+D_1-D_2)+a_1R_{sz2}(\tau+D_1)+$$
$$+a_2R_{sz2}(\tau-D_2)+R_{z1z2}(\tau) \tag{3}$$

The first component of the function (3) is the autocorrelation function of the original signal, which is maximum for the argument 0. Accordingly, the delay between the recorded signals is the argument of the maximum of the crosscorrelation function, expressed as in relation (4).

$$\Delta t = D_2 - D_1 = \arg(\max(R_{r1r2}(\tau))) \tag{4}$$

The crosscorrelation function may be computed as in relation (5), where n is the length of the recorded sequence, or by using the direct and inverse DFT of the recorded signals.

$$R_{r1r2}(\tau) = \begin{cases} \frac{1}{n}\sum_{k=0}^{n-1} r_1(t+k)\cdot r_2(t+k+\tau), & \tau \geq 0 \\ R_{r2r1}(-\tau), & \tau < 0 \end{cases} \tag{5}$$

Fig. 8 presents a record of two audio signals (an 800 samples sequence only) and their crosscorrelation function (the argument in the figure is expressed as an integer, it will be multiplied by the sampling period).

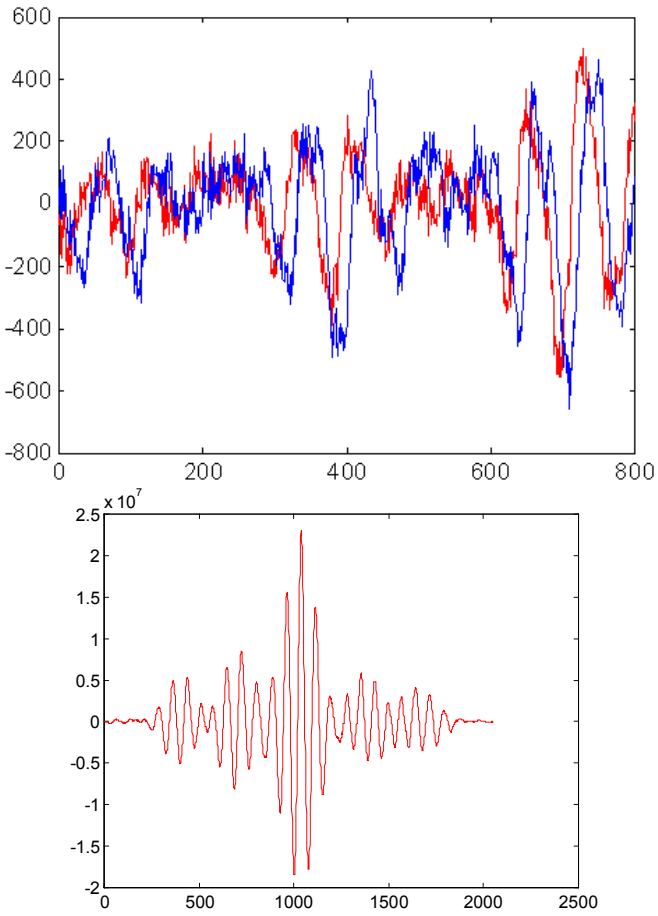

Figure 8. The recorded signals and their crosscorrelation function.

A distinct problem, affecting the resolution of the position measurement, is the discrete nature of the time. The maximum value of the delay measured through crosscorrelation and used in relation (1) is 0.21m / 340 m/s = 620 μs, i.e. less than 26 sampling periods. In fact, the resolution is not uniform on this interval, it is very poor when DoA approaches 90 degrees. In order to obtain reliable measurements, the delay is not directly measured through (4). The maximum of the function (3) and 4 other

neighbours are fitted to a 2nd degree polynomial and the argument of the maximum of this polynomial is used instead of (4). This method allows to use fractional values of the delay, more close to the real value. The average time error compensated through the fractional value is a quarter of sampling period, which means compensated angle errors of up to 1.5 degrees.

B. Position measuring (source localization)

Assuming the positions of the nodes are known (including the orientation of the microphone array), the DoA of the sound at each node is enough for determining the position of the source. The parameters x_k, y_k, θ_k stand for the position of the node and the orientation of the microphone array, in the horizontal plane. If using only two receptors, the position of the audio source is determined by the relations (6):

$$x_s = \frac{y_1 - y_2 + x_2 \mathrm{tg}\beta_2 - x_1 \mathrm{tg}\beta_1}{\mathrm{tg}\beta_2 - \mathrm{tg}\beta_1}$$

$$y_s = \frac{y_1 \mathrm{tg}\beta_2 - y_2 \mathrm{tg}\beta_1 + (x_2 - x_1) \cdot \mathrm{tg}\beta_2 \cdot \mathrm{tg}\beta_1}{\mathrm{tg}\beta_2 - \mathrm{tg}\beta_1} \quad (6)$$

where the variable $\beta = \theta + \alpha$.

The relations (6) may be reformulated as in (7), in order to avoid large errors, when one of the angles tends to 90 degrees.

$$x_s = \frac{(y_1 - y_2) \cdot \cos\beta_2 \cdot \cos\beta_1 + x_2 \cdot \sin\beta_2 \cdot \cos\beta_1}{\sin(\beta_2 - \beta_1)} -$$

$$- \frac{x_1 \cdot \sin\beta_1 \cdot \cos\beta_2)}{\sin(\beta_2 - \beta_1)}$$

$$y_s = \frac{y_1 \cdot \sin\beta_2 \cdot \cos\beta_1 - y_2 \cdot \sin\beta_1 \cdot \cos\beta_2}{\sin(\beta_2 - \beta_1)} +$$

$$+ \frac{(x_2 - x_1) \cdot \sin\beta_2 \cdot \sin\beta_1}{\sin(\beta_2 - \beta_1)} \quad (7)$$

The case of 2 receptors is very simple, but is subject to large errors, as the position of the source approaches the line crossing the two nodes (the denominator of the relations (7) tends to 0). This method is reliable only when the position of the audio source is limited to angles below 60 degrees, with respect to both receptors. In order to overcome this drawback, at least 3 nodes have to be involved, as illustrated in figure 1. For simplicity, this is the case to be presented, but it can be easily extended to more than 3 nodes.

The values of DoA, determined at each node, lead to a system of 3 equations with two unknowns, which has no solution (in the general case). Therefore, the system is solved in the sense of the least square errors. This means the solution is the pair x_s, y_s, minimizing an error criterion on the set of the measured DoAs. The relation determining the variable β is (8):

$$\beta_k = \theta_k + \alpha_k = \begin{cases} \arccos\left(\frac{x_s - x_k}{\sqrt{(y_s - y_k)^2 + (x_s - x_k)^2}}\right), y_s > y_k \\ -\arccos\left(\frac{x_s - x_k}{\sqrt{(y_s - y_k)^2 + (x_s - x_k)^2}}\right), y_s < y_k \end{cases} \quad (8)$$

The chosen criterion is the sum of the square errors affecting the DoA, as in relation (9):

$$J(x_s, y_s) = \sum_{k=1}^{n} (\beta_k(x_s, y_s) - \theta_k - \alpha_k)^2 \quad (9)$$

where n stands for the number of nodes (there are 3 nodes in this experiment). The function (9) is continuous, differentiable and defined on all the area perceived by the nodes. The exception appears when the audio source crosses the line of the microphones (not in front of the node), which is out of the area of this application. Then, the solution is a pair satisfying the relation (10):

$$\begin{cases} \frac{\partial J}{\partial x_s} = 0 \\ \frac{\partial J}{\partial y_s} = 0 \end{cases} \quad (10)$$

Because of the nonlinearity of the functions (8), relation (10) is not proper for finding an analytical solution. A numerical iterative procedure is used instead. As usually, the procedure converges faster if the initial guess is close enough to the real position of the source. The algorithm follows the steps:

- all audio boards detect the audio event and record the sound arriving at 2 microphones
- the DoA is computed at each node, on the audio board (relation (1))
- the values of DoA are transmitted to the central computer
- the central computer measures the values of x, y, by minimizing the function (9).

C. Calibration

The calibration problem relies to the precision we know the positions of the nodes, i.e. the real values of the parameters x_k, y_k, θ_k. When measuring these parameters is not possible, a calibration stage has to be performed. For this purpose, some audio events are produced in known positions, i.e. in the plane of the microphone arrays, at known coordinates x_j, y_j. At least 4 such events are necessary, in order to minimize the effect of the noise. The parameters are computed in a minimization procedure, using the criterion (11):

$$J = \sum_{k=1}^{n} \sum_{j=1}^{p} (\beta_{kj}(x_k, y_k) - \theta_k - \alpha_{kj})^2 \quad (11)$$

where k is the index of the node and j is the index of the experiment (n nodes and p experiments). The function (11) can be rewritten as:

$$J = \sum_{j=1}^{p} (\beta_{1j}(x_1, y_1) - \theta_1 - \alpha_{1j})^2 +$$

$$+ \sum_{j=1}^{p} (\beta_{2j}(x_2, y_2) - \theta_2 - \alpha_{2j})^2 + .. \quad (12)$$

and it becomes obvious that each sum corresponds to a single node. This means the problem of minimizing (11) can be reduced to minimizing n separate functions. For each node, the function to be minimized is expressed as (dropping the index of the node):

$$J(x,y,\theta)=\sum_{j=1}^{p}(\beta_j(x,y)-\theta-\alpha_j)^2 \quad (13)$$

The algorithm follows the steps:

- the positions of the nodes are roughly estimated (they will be used as first approximation in the iterative procedure)
- at least 4 audio events are generated, in known positions, detectable by all nodes
- each node computes the DoA for all events
- the values of DoA are transmitted to the central computer
- the parameters (positions of the nodes) are determined in the central computer, through the iterative procedure that solves *n* equations similar to (13).

IV. Experimental Results And Conclusions

More series of tests were carried out, using the described network of embedded devices. The scene was a room of 8 x 6 meters and 3 nodes were used. The capture on audio boards was triggered by the level of the sound (10% of the maximum level. At each step, two sequences of 1024 samples were recorded (corresponding to the two microphones) and stored as integers (2 bytes). The crosscorrelation function was computed on BF561, in 5.3 ms, then the result was sent by BF537 to the central computer, via the Ethernet link. The computation of the position of the source took a negligeable time on a Pentium IV. The total necessary time for computing the position takes 25ms for data acquisition, 6ms for DoA computation, 10ms (average) for data transmission to the central computer, i.e. approx. 40 ms (the time for the other tasks was neglected). The time required by the calibration procedure was not considered, because it happens only once, off-line.

For DoA no larger than 70 degrees and distances no shorter than 1m, the position error was within the limits of 0.1m. The main cause of the error was the multipath propagation of the sound.

The variable position of the audio source was not a problem for the localization system. For an usual moving source, at 0.5m/s, the space covered during the position computation is about 20mm, considerable lower than the error limit. This result allows the sensor network to equally localize still and moving audio sources.

The audio source localization becomes more complicated, when more than one source is present at a time, when the level of the noise is increased and when multiple reflections occur. These cases require more analysis, in order to be solved.

The embedded devices described in this paper proved to be well suited for the localization problem. They compute and communicate fast, reducing the task of the central computer to a short minimization problem.

The resources of the node are almost unused in this application. Only the audio processing board is involved, and both the memory and the processing capacity are used at less that 0.1%. The resources required by more complicated tasks, such as beamforming for source separation, are still less than the resources available on the audio board. This shows that the embedded devices are capable of solving more challenging applications.

Further applications, using the presented embedded devices will be the audio source separation, audio events recognition, video localization and threats recognition. They are all suited for using a network of such devices.

Acknowledgment

The work described in this paper was partly supported by the EU FP6 grant 033279 (SENSE).

References

[1] Y. Huang, J. Benesty, J. Chen, Acoustic MIMO Signal Processing. Springer, 2006

[2] G. Zucker, L. Frangu, "Smart Nodes for Semantic Analysis of Visual and Aural Data", Proc. 5-th Int. Conf. Ind. Informatics (INDIN 2007), pp. 1027-1032, Vienna

[3] L. Frangu, M. Măzărel, C. Chiculiță, S. Epure, "An Embedded Platform for Smart Multiple Sensor Network", Proc. 3rd Int. Conf. "From Scientific Computing to Computational Engineering", July 2008, Athens

[4] J. Simo, G. Benet, G. Andreu, "Embedded Video Processing for Distributed Intelligent Sensor Networks", Proc. 3rd Int. Conf. "From Scientific Computing to Computational Engineering", July 2008, Athens

[5] D. Bruckner, R. Velik, G. Zucker, "Network of Cooperating Smart Sensors for Global-View Generation in Surveillance Applications" Proc. 6-th Int. Conf. Ind. Informatics (INDIN 2008)