

I-SENSE: Intelligent Embedded Multi-Sensor Fusion

Andreas Klausner¹, Bernhard Rinner¹, and Allan Tengg¹

¹Institute for Technical Informatics

Graz University of Technology

Graz, AUSTRIA

{klausner, rinner, tengg}@iti.tugraz.at

Abstract — *I-SENSE demonstrates the potential of combining the scientific research areas multi-sensor data fusion and pervasive embedded computing. The main idea is to provide a generic architecture which supports a distributed online data fusion on an embedded system. Due to their high onboard processing and communication power our proposed architecture is designed to perform sophisticated data fusion tasks in realtime. Another goal of I-SENSE is to dynamically change the configuration, thus, to be able to react to changes in the systems environment. This paper describes ongoing work in developing necessary hard- and software components in order to perform realtime multi-level data fusion. We present the distributed I-SENSE platform and introduce our multi-level fusion framework. First experimental results on embedded image fusion demonstrates the feasibility of our approach.*

Keywords: *sensor data fusion; multi-level fusion; distributed embedded systems; pervasive computing; traffic surveillance*

1 Introduction

Currently there is a strong trend towards integration of sensor, computing and communication technology into everyday life. The ultimate goal here is to provide as much support as possible while concealing the computing devices from the users. This trend is reflected in areas such as embedded computing, pervasive computing, intelligent infrastructures, smart sensors and ambient intelligence.

In this project we develop an intelligent multi-sensor fusion framework (I-SENSE) for embedded online data fusion. This multi-sensor fusion framework is targeted at various applications, such as intelligent infrastructures, pervasive computing and monitoring. The I-SENSE project, however, is applied to traffic surveillance as case study.

Fusing data from various sensors helps to improve the robustness and confidence, to extend the spatial and temporal coverage as well as to reduce ambiguity and uncertainty of the processed sensor data. In the I-SENSE project we exploit these characteristics

to improve the quality of traffic surveillance. Since current traffic surveillance systems are primarily based on video, integration of data from audio, infrared, supersonic and inductive loop sensors, helps to improve various parameters such as recognition rates, detection times, robustness and quality of service [1, 2, 3].

The major goal of this research is to investigate and develop a scalable and embedded architecture for various multi-sensor applications. Therefore, the I-SENSE framework is based on embedded *intelligent* sensor nodes with sufficient computing and communication performance and a suitable embedded architecture, which allows to distribute SW tasks among geographically distinct sensor nodes. The research is focused on (i) the development of the distributed embedded scalable architecture to allow dynamic reconfiguration as well as effective online optimization (ii) and the development of a SW framework to allow that high-performance sensor nodes perform sensory data fusion among geographically distinct sensor nodes.

By delegating the CPU-expensive data fusion tasks into the sensor nodes, the requirements concerning the communication bandwidth can be reduced compared to a centralized data fusion architecture. This makes widespread data-fusion applications more feasible. A light-weight middleware supports data-oriented communication services such that fusion tasks can easily communicate regardless of their mapping - either on the same, on a different CPU or even on a different sensor node. This framework maintains a global clock which can be accessed easily by all nodes in the system.

Multi-level data fusion is accomplished among three different levels of detail according to the amount of information they provide [4, 5]. The most basic level involves the fusion of multi-sensor data to determine the position, velocity, and identity of a tracking object. At this level, however, only raw, uncorrelated data is provided to the user, and therefore this level is called “raw-data fusion”. In comparison, level two data fusion provides a higher level of inference and delivers additional interpretive meaning suggested from the raw data and data will be fused on feature level. Therefore, this level is called “feature-level fusion”. Level three data fusion is designed to make assessments and provide recommendations to the user or human observer, much as occurs in model-based expert systems and is therefore called “decision fusion”.

The remainder of this paper is organized as follows: Section 2 sketches related and preliminary work. Section 3 presents the hardware and software of our I-SENSE approach. Section 3.1 discusses the two level architecture and the overall functionality of the I-SENSE project. Section 3.2 describes the HW/SW architecture as well as its configuration and reconfiguration capabilities. Section 3.3 presents the SW framework to perform distributed multi-level multi-data fusion. Section 4 shows the current status of our work in HW/SW implementation and traffic surveillance case study. Section 5 concludes the paper with a summary and an outlook on future work.

2 Related Work

A lot of research has been conducted over the last decades in sensor fusion and therefore data fusion has been given much attention in the engineering literature. Several data fusion algorithms have been developed and applied, individually and in combination, providing users with various levels of informational detail.

The key scientific problems can be assigned to the three fusion levels as shown below:

Raw-data fusion: The key problems which have to be solved at this level of data abstraction can be referred to (i) *data association* and (ii) *positional estimation*. Data association is a general method of combining multi-sensor data by correlation of one sensor observation set with another set of observation [6, 7, 5]. Common techniques for solving the positional estimation problem are focused on Kalman filtering and Bayesian methods and are discussed in [8, 9, 10].

Feature fusion: These approaches are typically addressed by (i) *Bayesian Theory* and (ii) *Dempster-Shafer Theory*. Bayesian Theory is used to generate a probabilistic model of uncertain system states by consolidating and interpreting overlapping data provided by several sensors [11, 12, 13, 14]. It also determines conditional probabilities from a priori evidence; these revised probabilities are called “a posteriori probabilities”. Bayesian theory is limited in its ability to handle uncertainty in sensor data. This can hinder the application of this data fusion technique because sensor data are by nature highly uncertain. Therefore, Dempster-Shafer theory is a generalization of Bayes reasoning that offers a way to combine uncertain information from disparate sensor sources. Further information can be found in [15, 16, 17].

Decision fusion: Fusion at the decision level combines the decisions of independent sensor detection/classification paths by Boolean operators or by a heuristic score (e.g., M-of-N, maximum vote or weighted sum). The two basic methods for making classification decisions are hard decisions (single, optimum choice) and soft decision in which decision uncertainty in each sensor chain is maintained and combined with a composite measure of uncertainty. There are a few studies published of level three data fusion in [18, 19, 20].

“Project Correlation”, funded by the U.S. Air Force, was the first approach to step back from the many application-specific and system-specific solutions and developed a set of generic/reusable engineering guidelines for an effective data fusion-problem solution. A methodology for fusion software development, based on the C4ISR architecture [21] is given. However, this architecture has too much overhead and is, therefore, not suitable for embedded systems.

A project that has on first sight quite many similarities with our project is called *DFuse* [22]. This research focuses on challenges of data fusion applications in wireless ad hoc sensor networks. *DFuse* is designed to be used on “motest” [23]. Available motest are not yet powerful enough to support *DFuse*, so iPaqs have been used instead. Both, the communication range and the communication bandwidth, is very limited between fusion-nodes. Furthermore, this framework suffers from lacking of “intelligent” sensors and is not providing approaches for multi-level fusion.

Over the last years our institute has designed and implemented high-performance embedded cameras that combine video sensing, video processing and communication within a single embedded device. Our smart cameras [24, 25, 26] are designed as reconfigurable and flexible processing nodes with the capability to perform high-level tasks onboard. Networks of distributed smart cameras are an emerging technology for a broad range of applications, including smart rooms, intelligent infrastructure, tracking and motion analysis.

3 The I-SENSE Approach

In this section we describe the I-SENSE project in more detail. The description is organized in the following three parts: (i) the distributed embedded platform, (ii) the HW/SW architecture, and (iii) the multi-level fusion framework.

3.1 I-SENSE: Distributed Embedded Platform

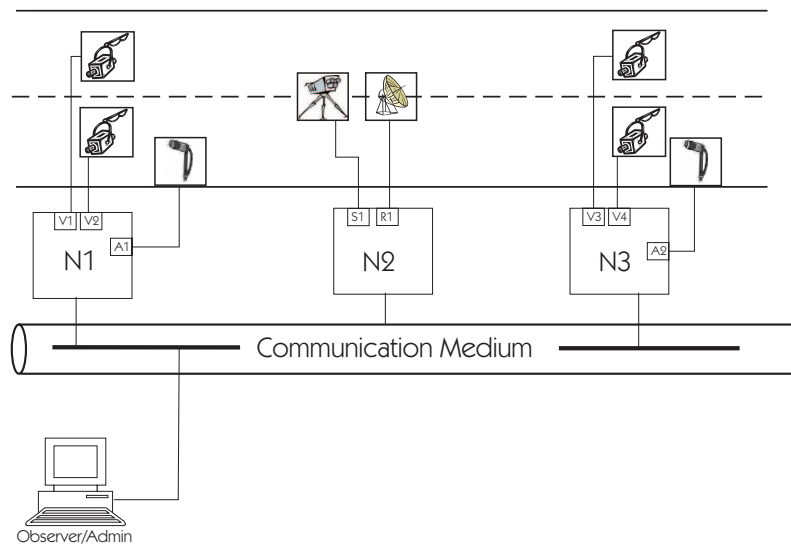


Figure 1: The I-SENSE platform as a scalable, distributed embedded system. ($N1 \dots N3$: Sensor nodes, $V1 \dots V4$: Video interfaces, $A1 \dots A2$: Audio ports, $R1$: Radar interface and $S1$: Spectral imaging interface)

The I-SENSE platform consists of a two-level architecture which is derived from our SmartCam architecture (cp. Section 2). The top-level is composed of a network of geographically distributed sensor nodes which are connected via a common communication medium (cp. Figure 1). We use primarily standard wired Ethernet as communication medium but also wireless communication such as GPRS or wireless LAN is supported.

The sensor nodes themselves represent the bottom-level of our I-SENSE platform and are the main processing components of our I-SENSE platform, e.g., all sensor fusion processing and system management takes place on these components.

The scalability is supported in several ways. First, the number and type of sensors can be adapted at the individual sensor nodes. Second, the processing and communication performance of each sensor node can be easily modified. Third, the number of sensor nodes can be easily adapted in our I-SENSE architecture.

The prototype of the I-SENSE sensor node uses components-off-the-shelf hardware in order to test and evaluate the distributed fusion system (cp. Figure 2). This prototype serves as feasibility demonstration of our approach. It provides the required computing and communication performance. More details are shown in Section 4.

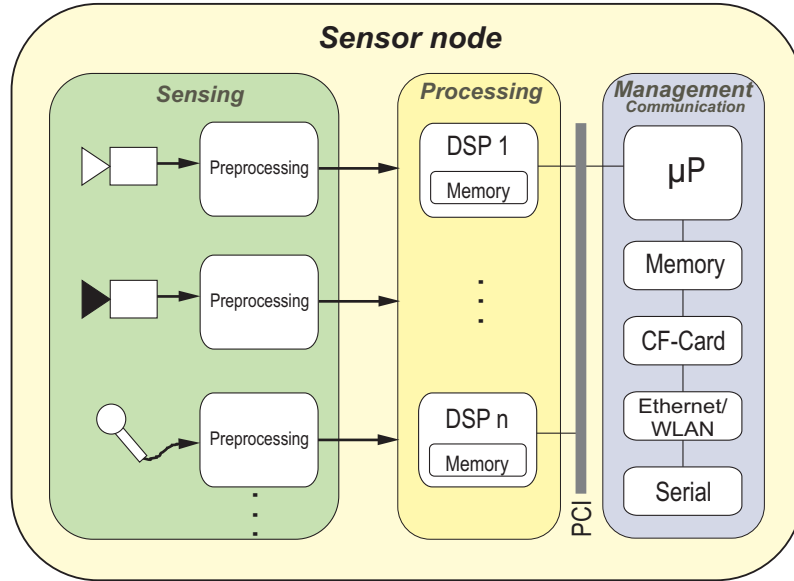


Figure 2: The structure of the sensor node

3.2 I-SENSE: HW/SW Architecture

In order to develop a generic HW/SW architecture for embedded data fusion we've identified the following challenges: (i) simple and expressive specification of the configuration problem, (ii) effective online optimization, (iii) lightweight runtime environment on the distributed embedded platform supporting efficient synthesis, and (iv) synchronization of intra- and inter-node sensor data. The first three challenges are general for distributed embedded systems whereas the last one is specific for distributed data fusion.

Two models are maintained for the specification of the configuration problem: The "hardware model" and the "software model". Figure 3 shows our online configuration method of the proposed fusion architecture and includes examples for the two models. The structural functional blocks are described in the following:

Fusion Model: The *Fusion Model* specifies the functionality of the fusion application.

It defines all sensor nodes, fusion nodes and their interaction among each other in form of a fusion tree. Three different types of nodes are possible:

Sensor Interfaces ($S1 \dots S7$) always build the bottom layer of the fusion tree, since they acquire the data from the environment directly and independently from other nodes. Each sensor node basically consists of a hardware interface and a driver to support the device at the operating system level. Data is acquired and converted into a suitable format, timestamped correctly and transferred to the connected fusion nodes.

Fusion Nodes ($F1, F2, F9, F10, F11, F12$) in contrast fully depend on data from other fusion nodes or sensor nodes to produce an output. Their main task is to wait until new input data becomes available, ensure the temporal alignment, calculate the output vector and distribute it to all connected nodes.

Filter Nodes ($F3, F4, F5, F6, F7$) are functional identical with fusion nodes except

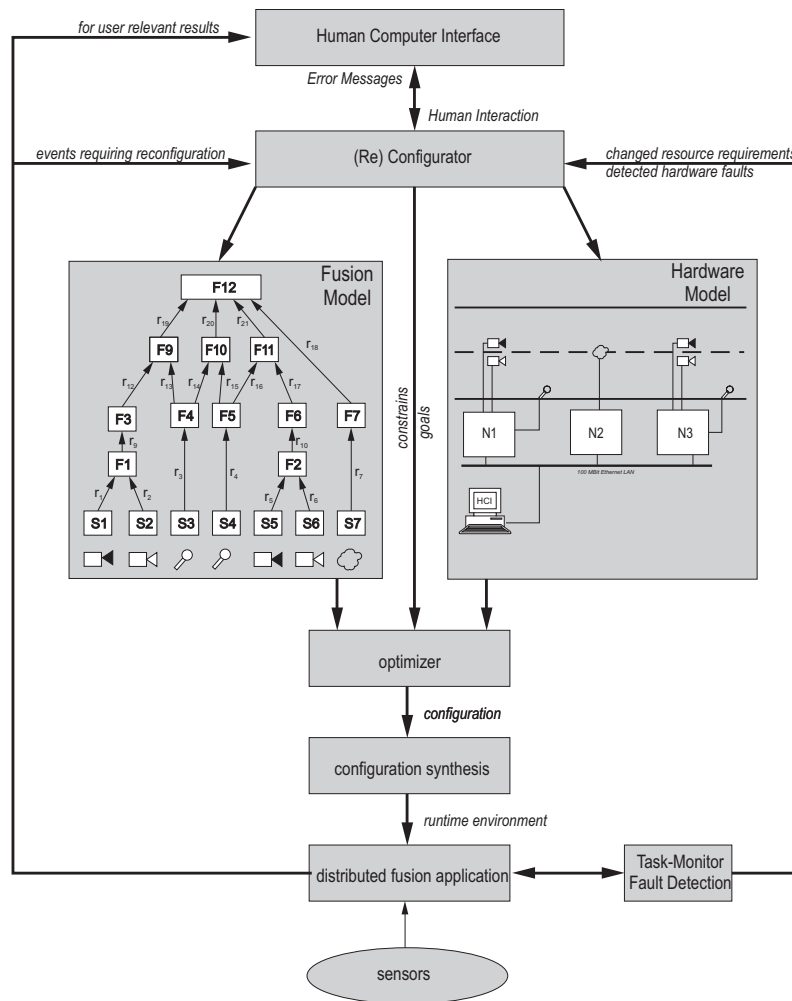


Figure 3: The Configuration method

they have only one input channel. Therefore, they do not have to care about temporal alignment. Splitting up a complex fusion node into many smaller fusion- and filter nodes is quite often a good idea, since smaller nodes can be placed better on the distributed embedded system and can be executed potentially parallel on different processors.

The proposed architecture requires for each fusion-, filter- and sensor-node an implementation in form of a relocatable library for at least one processor type in use. Additionally to the implementation, the worst case requirements of hardware (CPU cycles, internal and external memory, DMA channels, ...) as well as the communication bandwidth of each node in the fusion model must be specified for every implementation.

Hardware Model: The *Hardware Model* describes the distributed embedded system where the fusion application should run. It basically consists of a set of connected hardware nodes (N1 ... N3) with specific parameters (computing power, size of memory, different sensors). We provide a module which explores the embedded system au-

tomatically. This has two advantages: (i) faulty or missing hardware nodes can be found during start up and (ii) the “hardware model” can be parameterized during the initialization process.

Optimizer: Both models are used as input to an *Optimizer* which computes the optimal mapping of the real-time fusion tasks onto the sensor nodes, subjected to some constraints. Various optimization goals can be specified such as minimizing power consumption or minimizing the communication. Constraints help to enforce the mapping of a individual fusion tasks onto a dedicated processor. We plan to utilize a genetic algorithm for this optimization.

Configuration Synthesizer: The *Configuration Synthesizer* integrates the fusion tasks into the runtime environment of the distributed embedded platform. This involves three major steps: First, the dynamic link libraries for the fusion nodes have to be loaded on the specified hardware node. In the second step the defined communication channels have to be established between the fusion tasks. In the last step the initialization routine of all sensor- and fusion nodes are called. They are expected to initialize the fusion tasks and their required hardware.

Task Monitor: The *Task Monitor* runs on every sensor node of the embedded system. It checks periodically the health of its processor, the communication links and the utilization of the resources under its administration. If tasks use more resources than the programmer declared in advance, the reconfigurator is notified. On the other hand, resources that are claimed by a task but never used are reported too.

(Re)Configurator: The *(Re)Configurator* is responsible for maintaining the fusion- and hardware-model. Either the user or the system itself can request a change in one of those models. The user specifies rather the large scale functionality by selecting one of many available fusion trees doing different jobs. In the hardware model the user might disable or add a node.

Optimization and (re-)allocation may not only be triggered at start time but also during runtime. There are three reasons why the reconfigurator is invoked autonomously from the system: The most likely scenario is that a rather high fusion node detects a relevant event and decides to adapt the fusion tree to better monitor this event. Other less likely scenarios requesting a reconfiguration are hardware faults and suspicious software tasks. In case of hardware faults, the faulty device is removed from the hardware model and the optimizer is called again. If it is somehow possible, the optimizer will find an alternative mapping without the faulty node and the fusion application will continue to work unrestricted. If the faulty node is irreplaceable, the system can continue its work only in a restricted manner. If a software task allocates more resources than it was declared before, the framework will tolerate this as long as the resources are available and update the “fusion model” to the real requirements to prevent future conflicts.

Another very important requirement for our generic framework is the synchronization of the data delivered by different sensors. We plan to implement common methods for clock synchronization in the framework as described in [27, 28]. The fusion algorithms can then perform temporal alignment on data packets either from the same or distinct sensor nodes.

3.3 I-SENSE: Multi-Level Fusion Framework

As the name multi-sensor data fusion implies, it is a technique by which data from several sensors are combined through a data processor to provide comprehensive and accurate information. Although the provision of a single data stream from multiple inputs is advantageous, the powerful potential of this technology stems from its ability to track changing conditions and anticipate impacts more consistently than could traditionally be done with a single data source. Thus, our multi-sensor data fusion framework makes it possible to create a synergistic process in which the consolidation of individual data creates a combined resource with a productive value greater than the sum of its parts.

To achieve this aim our approach to perform multi-level data fusion is to combine data from the different sensors at two stages, as shown in Figure 1. First, *intra-node* fusion takes place at a single sensor node where raw sensor data or abstracted features are combined. Second, *inter-node* fusion combines abstracted data from various geographically distinct sensor nodes. This leads to an aggregation of the advantages of individual abstraction levels. Each jump between data fusion levels represents a corresponding leap in technological complexity to produce increasingly valuable informational detail.

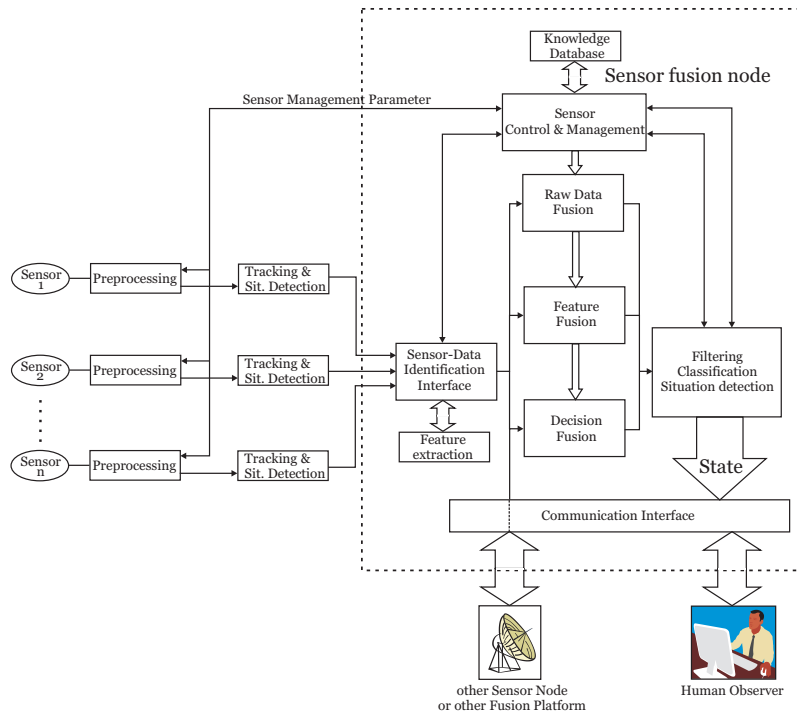


Figure 4: The multi-level Data Fusion framework

The multi-level data fusion framework is presented in Figure 4. The three fusion methods (i) raw-data fusion, (ii) feature-level fusion and (iii) decision fusion are the heart of this SW framework. The output of these fusion methods are then combined to derive the current state of the fusion node. Typical methods for deriving this state are filtering, classification and situation detection. The state of the fusion node is sent to other sensor nodes or a visualization node via the communication interface. As indicated in this

figure, most of the required computation takes place on the fusion nodes; only preprocessing, simple object tracking and raw data based situation detection is assigned to the smart sensors. The two bottom level fusion methods combine data from sensor directly attached to the fusion node; the top level method uses state information from remote fusion nodes. In order to realize a flexible solution our multi-level fusion is based on the following functionalities:

Raw-Data Level Fusion: Each sensor performs a single-source positional estimate in the sensor state space. These estimates are then combined to an aggregate estimate. Since raw-data fusion is performed in the raw data space of the sensor, it can only be applied for similar sensor types (e.g. visual sensor & infrared visual sensor).

Feature Level Fusion: Each sensor performs a single-source positional estimation, producing a state vector from each sensor. This means, each sensor provides an estimate of the position and velocity of an object (cars, cargo, persons, etc.) or an observed situation, based only on its own single source data. These estimates of position and velocity or observed situations are input to a data fusion process to achieve a joint state vector estimate based on multiple sensors. The functions of data alignment and association/correlation are still performed at the state vector level. In addition, the association/correlation process is conceptually easier than performed for raw data level fusion. To meet this need we will use adapted and enhanced “Dempster-Shafer” data fusion process.

Decision Fusion: At this level the derived states of different sensor nodes are combined. A state of a sensor node corresponds to a high-level assessment of the sensor’s observed area. Examples for such an assessment in traffic surveillance include the detection of a traffic jam, the identification a specific vehicle, or the detection of lost cargo. Both, the assessment of states as well as their combination requires an interpretation of the lower level fusion outputs and a lot of domain knowledge. In our research project we investigate two methods for combining the states. In the first method, the procedure for combining states is specified by domain-dependent rules. In the second method, we apply a classifier based on neural nets which is trained with previously recorded and classified traffic sequences.

In order to fulfill the constraint for an generic fusion framework, we will provide the basic fusion functionalities as mentioned above. A system administrator is initializing the framework by describing each sensor and the corresponding feature extraction algorithms. Therefore a GUI will be provided. During our research we will evaluate how generic feature extraction algorithms can be designed for a particular set of sensors and a defined application. The information about sensors and corresponding algorithms will be stored in a knowledge database. Fusing potentially conflicting results at the individual fusion levels is a key challenge. We will therefore evaluate different robust data abstraction methods and a synchronization model among the different fusion levels.

4 Implementation

Current work of the I-SENSE project is focused on integrating image calibration, registration and fusion into the I-SENSE platform to perform raw-data fusion of visual data.

Currently, a case study is conducted to evaluate the possibilities of integration high performance audio cards to our proposed system. This audio data is applied to traffic surveillance tasks. Feature extraction algorithms are designed and integrated into our system next.

A sensor node of our I-SENSE architecture, as shown in Figure 2, has been implemented with standard high-performance components. Therefore, the ePCI-101 Kontron board, together with a PCI backplane, serves as I-SENSE multisensor platform. This baseboard is equipped with an Intel Pentium M processor with passive heat sink running up to 1,6 GHz, 512 MB external memory and the current backplane offers four PCI slots. The ePCI-101 board provides two 100 MBit/sec Ethernet ports, two serial ports, several USB-ports, a VGA connector and IDE connectors. The on-board CF slot is well suited to store the operating system, the fusion software framework and the initial configuration on an affordable 256 MB flash card.

Network Video Development Kits (NVDK) from ATEME serve as the DSP platform, equipped with Texas Instruments TMS320C6416 DSPs running at 600 MHz and with a total of 264 MB of memory. The CMOS sensor KAC-9628 from Kodak, is used to capture color images. This image sensor provides a high-dynamic range of up to 110 dB at VGA resolution. To extend the visible spectrum an infrared camera with night-vision features is connected to the NVDK. A professional audio card (Audiophile 2496 from M-Audio) allows the system to capture audio signals with up to 96 kHz sampling rate and 24 bit resolution. Further sensors like inductive loop sensor or radar equipment will be connected via PCI or USB to our I-SENSE platform.

The performance of the I-SENSE architecture will be demonstrated and evaluated in a traffic surveillance case study. The case study is organized in two steps. First, we will setup an I-SENSE prototype in our lab and demonstrate the feasibility of our approach to traffic surveillance. For this demonstration we must emulate the sensor input to the I-SENSE prototype. This emulation can be realized either by play back previously recorded multimedia data from traffic scenarios or by integrating the “expected” sensor readings into the sensor nodes. Second, we will deploy the prototype in a real-world traffic surveillance application. We plan to integrate the I-SENSE platform in a monitoring station of highway or tunnel.

5 Conclusion

Our I-SENSE prototype is a major step towards the development of a fully embedded distributed realtime data fusion system. Instead of performing the computation on a central server we propose to delegate the functionality to our intelligent embedded sensor nodes. The system architecture becomes more flexible and scalable, the overall communication bandwidth is reduced and the entire multi-level data fusion system is able to autonomously and dynamically react to detected events. This novel architecture poses, however, strong requirements on the systems hard- and software. Recent technological advances support this trend towards embedded systems with powerful microprocessor architectures. Surveillance systems become more “sensitive” by combining visual, acoustic, tactile or location-based information. This leads to more accurate results and an extended spatial and temporal coverage as well as an increased robustness and confidence. As a key benefit for the user this will lead to increasing “positive detection rates” and decreasing

“false positive rates” in case of traffic surveillance.

From the perspective of potential users or application developers our architecture aims at (i) simplifying the specification and synthesis of data fusion applications, (ii) checking the feasibility whether the available hardware resources are sufficient for data fusion application, (iii) providing a generic architecture for data fusion.

Developing this I-SENSE prototype will give us insight that may be interesting for many other distributed embedded fusion applications as well. Our design is not specialized for a specific scope of application; the traffic surveillance application mentioned above is just an example. It could be used in a wide field of application, like robotics, medical systems, chemical processes. Robots usually have multiple sensors to navigate in their environment. In medical diagnosis systems it is often required to combine many independently created models of the body (X-Ray image, supersonic images, CAT, etc) to recognize a disease or an injury.

Further work will include (i) the implementation of advanced feature extraction algorithm based on multi-spectral video and audio data as well as inductive loop and light barriers, (ii) the implementation of our MATLAB based image calibration and registration for our I-SENSE platform, (iii) the upgrading of our current image-fusion into a real-time video-fusion, (iv) the implementation of our proposed multi-level fusion framework, (v) the evaluation of existing synchronization mechanisms and porting to the I-SENSE platform and (vi) the determination of a feasible allocation algorithm for the “Optimizer”.

References

- [1] C. S. Regazzoni, V. Ramesh, and G. L. Foresti. Introduction of the special Issue. *Proceedings of the IEEE*, 89(10), Oct 2001.
- [2] G. L. Foresti, C. Mähönen, and C. S. Regazzoni. *Multimedia video-based surveillance systems*. Kluwer Academic Publishers, 2000.
- [3] G. L. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis. Active Video-Based Surveillance System. *IEEE Signal Processing Magazine*, 22(2):25–37, March 2005.
- [4] R.J. Linn and D.L. Hall. A survey of multi-sensor data fusion systems. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 1470, pages 13–29, Orlando, FL., Apr. 1991.
- [5] J. Llinas and D. L. Hall. An introduction to multi-sensor data fusion. In *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, volume 6, pages 537–540, May–June 1998.
- [6] B. V. Dasarathy. More the merrier ... or is it? - sensor suite augmentation benefits assessment. In *Proceedings of the 3rd International Conference on Information Fusion*, volume 2, pages 20–25, Paris, France, Jul. 2000.
- [7] B. V. Dasarathy. Information fusion - what, where, why, when, and how? In *Proceedings of the 4th International Conference on Information Fusion*, volume 2, pages 75–76, Feb. 2001.
- [8] F. Daum. Book review on: Handbook of multisensor data fusion. *IEEE Aerospace and Electronic Systems Magazine*, 16(10):15–16, Oct. 2001.
- [9] J. Z. Sasiadek and P. Hartana. Sensor data fusion using kalman filter. In *Proceedings of the Third International Conference on Information Fusion*, volume 2, pages 19–25, Paris, France, Jul. 2000.
- [10] Q. Gan and C. J. Harris. Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronics*, 37(1):273–279, Jan. 2001.
- [11] A. Cameron and H.L. Wu. Identifying and localizing electrical components: A case study of adaptive goal-directed sensing. In *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pages 13–15, Arlington, VA., Aug. 1991.

- [12] P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, 1996.
- [13] K. Kim. Bayesian inference network: Applications to target tracking. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 1698, pages 360–371, Orlando, FL, Apr. 1992.
- [14] B. Moshiri, M. R. Asharif, and R. Hosein Nezhad. Pseudo information measure: A new concept for extension of bayesian fusion in robotic map building. *Information Fusion*, 3(1):51–68, 2002.
- [15] T. Payne. Central fusion of sensor information using reasoned feedback. *Complex Systems: From Biology to Computation*. IOS Press, pages 248–259, 1993.
- [16] F. Butini, V. Cappellini, and S. Fini. Remote sensing data fusion on intelligent terminals. *European Transactions on Telecommunications and Related Technologies*, pages 555–563, Nov.–Dec. 1992.
- [17] V. Koval, V. Turchenko, V. Kochan, A. Sachenko, and G. Markowsky. Smart license plate recognition system based on image processing using neural network. In *Proceedings of the Second IEEE International Workshop on Advanced Computing Systems: Technology and Applications, 2003*, pages 123–127, 2003.
- [18] P. Jackson. *Introduction to Expert Systems*. Addison Wesley, 3rd edition edition, Dec. 1998.
- [19] J. Liebowitz. Rule-based expert systems. In *The Electrical Engineering Handbook*. CRC Press, Boca Raton, FL, 1993.
- [20] V. Diehl, F. Shields, and A. Hauter. Testing of multi-sensor automatic target recognition and fusion systems. In *Proceedings of the 6th national Sensor Symposium, 1993*, volume 1, page 155, Aug. 1993.
- [21] C4ISR Architecture Working Group. Capstone C4ISR Architecture. <http://www.fas.org/irp/program/core/c4isr.htm>, 1998.
- [22] Rajnish Kumar, Matthew Wolenetz, Bikash Agarwalla, JunSuk Shin, Phillip Hutto, Arnab Paul, and Umakishore Ramachandran. DFuse: A Framework for Distributed Data Fusion. In *Proceedings 2003 ACM SensSys*, Los Angeles, CA., Nov 2003.
- [23] Kris Pister, Joe Kahn, and Bernhard Boser. *SMART DUST - Autonomous sensing and communication in a cubic millimetre*. <http://dnclab.berkeley.edu/nota/RunningMan/Mote.htm>, Berkeley, Ca., 1999.
- [24] Michael Bramberger, Roman Pflugfelder, Arnlod Maier, Bernhard Rinner, Helmut Schwabach, and Bernhard Strobl. A Smart Camera for Traffic Surveillance. In *Proceedings of the First Workshop on Intelligent Solutions for Embedded Systems (WISES 2003)*, Vienna, Austria, June 2003.
- [25] Andreas Doblender, Arnold Maier, Bernhard Rinner, and Helmut Schwabach. Improving Fault-Tolerance in Intelligent Video Surveillance by Monitoring, Diagnosis and Dynamic Reconfiguration. In *Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems*, pages 194–201, Hamburg, Germany, May 2005.
- [26] Michael Bramberger, Andreas Doblender, Arnold Maier, Bernhard Rinner, and Helmut Schwabach. Distributed Embedded Smart Cameras for Surveillance Applications. *Computer*, 39(2):68–75, February 2006.
- [27] L. Thiele, P. Blum, and L. Meier. Improved interval-based clock synchronization in sensor networks. In *3rd International Symposium on Information Processing in Sensor Networks, 2004*, Berkeley, USA, April 2004.
- [28] M. Horauer and R. Höller. Integration of high accurate Clock Synchronization into Ethernet-based Distributed Systems. In *Proceedings of the International Conference on Advances in Infrastructure for e-Business, e-Education 2002*, Italy, August 2002.