Online Multi-Criterion Optimization for Dynamic Power-Aware Camera Configuration in Distributed Embedded Surveillance Clusters

A. Maier, B. Rinner, W. Schriebl
Institute for Technical Informatics
Graz University of Technology
A-8010 Graz, Austria
{maier, rinner, schriebl}@iti.tugraz.at

H. Schwabach
Video and Safety Technology
ARC Seibersdorf research
A-2444 Seibersdorf, Austria
helmut.schwabach@arcs.ac.at

Abstract

Intelligent video surveillance (IVS) systems are based on the recent development of so called embedded smart cameras. Delivering a good service quality in IVS usually results in a higher level of computing activity and therefore in increased power consumption.

This work presents PoQoS, a novel approach that aims in maximizing the service quality (i.e., the number of IVS-services and their QoS) while minimizing the system's power consumption. PoQoS enables power-aware reconfiguration of services and hardware resources in distributed logical clusters of embedded smart cameras. In order to find optimal camera configurations during operation, PoQoS integrates PoSeGA, an online genetic multi-criterion optimization algorithm. A configuration manager properly selects among optimized camera configurations and consequently initializes intra-camera or intra-cluster power-aware reconfiguration with respect to application- and situation-specific context.

The evaluation of PoQoS on the PoQoCam, a power-efficient embedded smart camera platform, shows the feasibility of the presented approach.

1 Introduction

Intelligent video surveillance (IVS) systems are based on the recent development of so called smart cameras [11, 2]. These cameras combine video sensing and data analysis, video data compression as well as transmission in a single embedded device. Beside high demands in computing performance, power-awareness is also of major importance in IVS. Especially in solar-powered cameras it leads to prolonged operation time and smaller device sizes. Furthermore, recent technologies such as Power-over-Ethernet are also deployed in these embedded designs but do have strict limitations on the amount of available energy as well.

In traffic surveillance for instance, services like MPEG-video streaming, accident detection, vehicle classification or the computation of traffic statistics usually have high demands in Quality-of-Service (QoS). Typical QoS-parameters include frame rate, transfer delay, image resolution and video compression rate [10].

An IVS system may get partitioned into distributed logical groups of typically co-located smart cameras – named surveillance clusters. A so called cluster configuration is represented by a set of IVS-services in various QoS-levels executed in a surveillance cluster that is composed of a specific number of embedded smart cameras. The service quality of a configuration represents the number of executed services and its corresponding QoS-levels.

This paper presents a novel approach for online multi-criterion optimization (MCO) and autonomous dynamic power-aware reconfiguration of services within distributed logical clusters of embedded smart cameras.

The approach is mainly based on a genetic algorithm – named PoSeGA – that is specially tailored for its use on embedded smart cameras. PoSeGA performs online MCO for the two desired objectives 'maximizing the service quality' and 'minimizing the power consumption' for embedded smart cameras in a distributed surveillance cluster. It computes a set of Pareto-optimal camera configurations that represent different tradeoffs in service quality and the corresponding power consumption with respect to a given cost model for both desired objectives.

Our presented approach also includes permanent sensing and analysis of application- and situation-specific contextual information within the cluster. Its advantage is a more autonomous system behavior instead of using pre-defined modes for cluster reconfiguration. The contextual information then can get used for selecting a single proper configuration among the computed Pareto-set.

Our approach is evaluated by a power-efficient imple-

Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)
1550-445X/06 $20.00 © 2006 IEEE

mentation of an embedded smart camera, the PoQoCam. Experimental results show the feasibility of the presented genetic algorithm for its use on embedded smart camera platforms in IVS.

The remainder of this paper is organized as follows: Section 2 briefly describes relevant related work on multi-criterion optimization and power-aware embedded computing systems. Section 3 presents our novel approach for autonomous dynamic power-aware cluster reconfiguration with embedded smart cameras. In section 4, PoSeGA, a feasible implementation of a genetic algorithm for online optimization of camera configurations is described. In Section 5 we present the PoQoCam, a power-efficient implementation of an embedded smart camera which is used for the experimental results presented in Section 6. Section 7 finally concludes the paper and outlines future work.

2 Related Work

A lot of real-world optimization problems usually have more than one, often conflicting objectives for optimization and therefore are referred to as *multi-criterion optimization* problems [4].

Solving a MCO-problem does not result in a single scalar that represents an optimal value but in a set of several so called *non-dominated* solutions (also referred to as *Pareto-optimal* solutions). However, none of the Pareto-optimal solutions is 'better' than another one in general but only in at least one criterion.

So called evolutionary approaches – including genetic algorithms (GAs) – are used to solve MCO problems [8]. GAs are stochastic search algorithms that are based on the evolutionary ideas of natural selection and genetics. They are an applicable and robust approach especially for MCO problems with a large and complex search space.

Minimizing the power consumption of embedded computing systems is also an area of intense research. A commonly used online method for power-aware system reconfiguration is dynamic power management (DPM) [1]. DPM is based on the observation that a lot of power is wasted because of system components that are fully powered up even if they are not in use.

Power-aware distributed environments have been researched in several relevant related research projects. The work of [9] for instance investigates a power-aware multimedia streaming to heterogenous handheld devices. A unified framework for DPM of the CPU and memory has been implemented. Further power savings are achieved by graceful degradation of QoS. Similar to that, [3] researches the tradeoff between image quality and power consumption, whereas the work mainly focuses on sophisticated image compression techniques.

Genetic algorithms have recently also been used for the optimization of power-efficient device scheduling as summarized in [6]. In [5], a GA is used for the optimization of task-scheduling of low-power clients that are connected through a channel of limited bandwidth in a wireless embedded client-server system. However, while the presented approach covers a related concept, its application may not be feasible for online optimization in IVS due to its performance results.

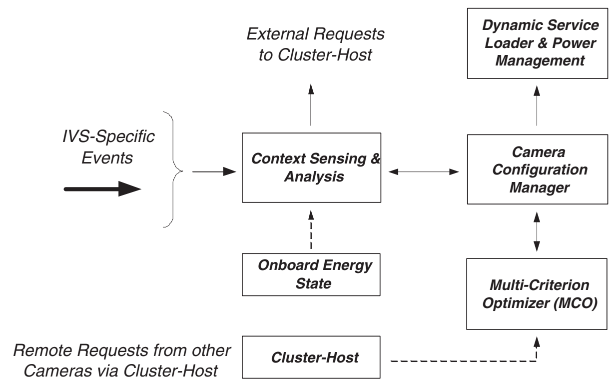


Figure 1. Basic PoQoS functionality.

3 An Improved Approach for Dynamic Power-Aware Cluster Reconfiguration

In [7], we introduced PoQoS, a hierarchical approach for QoS-driven dynamic power management in IVS systems. It focuses on the use of several pre-defined modes that are managed by a central hosting unit. The modes trigger the cluster-wide power- and service-configuration of the individual smart cameras and get altered for different application-specific demands including, e.g., human surveillance as well as alarm- or energy constrained- situations.

However, the requirements for the optimal power- and service-settings of a cluster configuration is application- and situation-specific and varies dynamically during operation. Therefore, the PoQoS approach is improved in this work in order to allow autonomous dynamic power-aware intra-camera and intra-cluster reconfiguration of services and hardware resources in distributed IVS systems consisting of embedded smart cameras. Fig. 1 shows the basic functionality of PoQoS on an embedded smart camera. The approach now features a unit for sensing and analyzing contextual information, an online multi-criterion optimizer as well as a configuration manager that triggers the onboard dynamic power manager. Furthermore, it includes mechanisms that allow handling of both in- and outgoing requests for the relocation of IVS-services within the cluster via a so called cluster-host.

3.1 Sensing and Analyzing Context in IVS

The contextual information in an IVS cluster is composed of numerous events that need to be sensed and analyzed. These events may be generated within the camera itself, within the cluster (i.e., a camera causes an event that gets sensed by other cameras), or by direct user interaction. Whenever an event occurs, a corresponding event-message is generated and distributed within the cluster. Since the types of events typically are application-specific, it makes sense to maintain an individual collection of event-messages in a configurable context-data- and rule-file in each camera.

In traffic surveillance for instance, the high QoS-level of a camera configuration needs to be kept in the context of an alarm situation (e.g., a stationary vehicle was detected) even if a low-energy event occurred. The context-file then may cover the following types of events: (1) the energy capacity of a camera is below (or again above) a pre-defined critical threshold value; (2) the size of the surveillance cluster is changed (e.g., when a camera is turned off); (3) the service quality of a camera is changed due to a pre-defined scheduled profile (e.g., at night); (4) direct user interaction (e.g., power-down of a camera); (5) events caused by IVS-services themselves (e.g., detection of an alarm generated by a stationary vehicle detection algorithm); (6) a request for relocation of camera configurations.

The above IVS-events normally have an impact on the service-level and power consumption of a smart camera. The analysis of these events provides knowledge about its contextual information. Since the information is brought into line with the current individual camera configurations, it allows making decisions about one of the following actions: (1) try to relocate services within the cluster; (2) keep the current camera configuration; (3) choose another (already pre-computed) configuration among the Pareto-set; or (4) re-trigger the optimizer due to a new search space.

3.2 Power-Aware Intra-Cluster Reconfiguration

PoQoS considers making a remote request for relocation of a service within the surveillance cluster due to, e.g., low energy in a camera. The remote requests are handled and dispatched by a so called cluster-host as illustrated in Fig. 2.

The cluster-host contains a unit that dispatches remote requests (RRD). It maintains a dynamically updated list that contains information about possible target cameras with enough free computing- and energy resources, whereas the list allows a search space reduction. If the relocation of services is feasible (i.e., due to no camera affinity of a service), the considered services get distributed among possible target cameras.

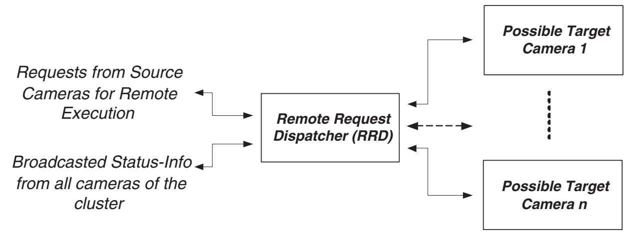


Figure 2. Basic features of the cluster-host.

Before relocation, a remote request triggers the multi-criterion optimizers in the target cameras by also sending a service's cost model. The optimizers on the target cameras then compute a Pareto-set with possible configurations containing the additional new service(s).

The cluster-host then chooses the target camera with the 'best' possible tradeoffs. After that, the services of the considered source camera are transferred to the selected target camera. The dynamic service- and power manager in the source camera then may power-down its freed hardware resources.

Since a key feature of smart cameras is their network connectivity, several different ways for relocating services can be implemented. For instance, it may include the use of a multi-agent system as described in [2].

As a matter of implementation, the cluster-host itself may be located on any smart camera within the cluster. It then makes sense to locate it on the camera with the most remaining energy resources whereas then it changes its location frequently.

3.3 Power-Aware Camera Reconfiguration

If a cluster-wide reconfiguration is not feasible due to the camera affinity of a service, reconfiguration is only allowed on camera-level.

The simplest way is to select a different configuration among an already computed Pareto-set. In that case, the contextual information needs to provide information about which power/service tradeoff is desired. In a low-energy context for instance, a camera configuration that utilizes less hardware resources is selected in order to allow the onboard dynamic service- and power manager to power-down the freed components.

4 Online Multi-Criterion Optimization of Camera Configurations

Since the application- and situation-specific demands for the service quality and power consumption are altered during operation, online optimization and dynamic reconfiguration of the cluster configuration is desirable.

A cluster configuration comprises of a set of individual camera configurations C_{Cam} that represent IVS-services in different QoS-levels on the given hardware resources $C_{Cam} = (Services \times QoS \times Resources)$. For a cluster of *c* cameras, the mapping of *s* services in up to q_s different QoS-levels on *r* hardware resources (i.e., CPUs) results in a theoretical upper bound of $(c \cdot r)^{s \cdot q_s}$ configurations that need to be considered during the optimization process. For *c=20, r=2, s=3* and $q_s = 5$ this would result in a search space with an upper bound of 40^{15} possible configurations that have to be addressed during optimization.

Since this may not be feasible for soft-realtime constraints in the given application, the optimizers search space has to be reduced. Therefore, PoQoS focuses on individual camera configurations separately as this typically results in a significantly smaller search space. Nevertheless, a reduced subset of the entire cluster search space is still considered by the functionality of the remote request dispatcher in cluster-host as described in the previous section.

We formulate the considered problem as combinatorial MCO problem that optimizes the costs of the camera configurations for the two objectives 'minimization of power consumption' and 'maximization of service quality' regarding to the three decision variables r, s and q_s. The individual cost functions for both objectives are given by equations 1 and 2:

$$Costs_P = \sum_{i=0}^{r} Costs_P(Resource_i) \cdot \mu(i) \quad (1)$$

$$Costs_S = \sum_{i=0}^{s} Costs_S(Service_i) \cdot \lambda(q_s(i)) \quad (2)$$

In Equ. 1, r is the number of the affected hardware resources and $\mu(i)$ is an individual weighting function for the actual utilization of each hardware component.

In Equ. 2, s is the number of the considered services (i.e., algorithms) and $\lambda(q_s(i))$ is a weighting function for each QoS-level q_s that is currently executed.

4.1 A Genetic Optimization Algorithm for IVS

It is the optimizer's task to compute a set of Pareto-optimal configurations. We have therefore, developed PoSeGA, a genetic algorithm that is suitable for solving the considered MCO problem online on embedded smart cameras. PoSeGA computes a set of non-dominated solutions that represent Pareto-optimal camera configurations by using the following input data: (1) p_{init}, the size of the initial population; (2) r, the number of the camera's hardware resources (i.e., CPUs); (3) s, the number of desired services for the camera; (4) cp, the cost model for the power consumption each task causes; (5) cs, cost model for the service quality levels of each task; (6) m, the probability for individuals mutating; and (7) c, the probability for individuals crossing over.

PoSeGA initializes a random population of an adjustable size p_{init}. Before each iteration, the generated population is checked if so called invalid individuals exist. For instance, if the algorithm generates an individual that would result in an CPU-load above 100%, it gets removed. Therefore, a new population is generated only by valid individuals. Non-dominated individuals are selected using a fitness function for both objectives. The basic functionality of PoSeGA is described by the pseudo code listed in Fig. 3.

```
Init_Population (p_init, r, s, q_s);
REPEAT
   Mutate (Population, m);
   Crossover (Population, c);
   Test_Fitness (Population, cp, cs);
   Remove_Dominated (Population);
UNTIL Convergence(Population);
```

Figure 3. Pseudo code of PoSeGA.

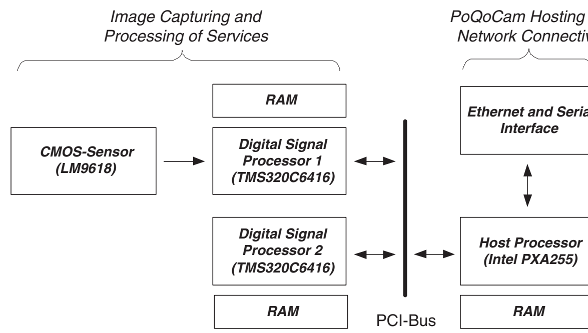


Figure 4. PoQoCam hardware architecture.

5 Implementation

We have implemented the PoSeGA algorithm on an embedded smart camera, the PoQoCam. The PoQoCam is a more power-aware variant of the *SmartCam* [2]. Fig. 4 shows the hardware architecture of the platform.

Fig. 5 shows the prototype implementation of the PoQoCam. It is based on a processor evaluation board from Kontron equipped with an Intel PXA255 processor that allows the dynamic adjustment of its processor speed to up to

Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)
1550-445X/06 $20.00 © 2006 IEEE

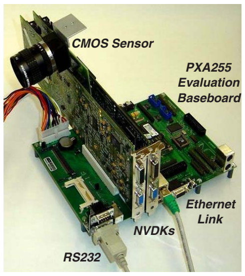


Figure 5. The prototype of the PoQoCam.

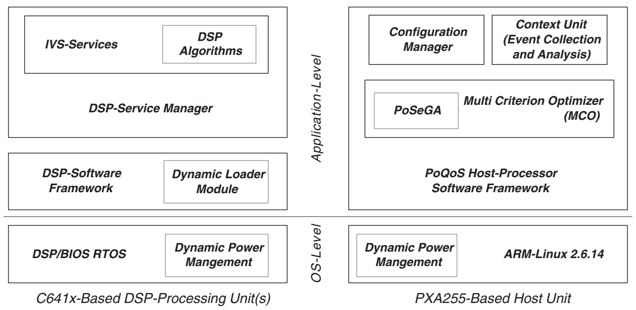


Figure 6. The PoQoCam software-framework.

400MHz. The prototype contains two DSP-cards (ATEME NVDK each comprising a C6416-DSP from Texas Instruments running at 600MHz), whereas a VGA-CMOS sensor is connected to one DSP. The interprocessor communication is build upon a PCI-bus (Rev. 2.1.). Fig. 6 shows the main components of the software-framework of the PoQoCam. It is basically similar to the software-framework implementation of the *SmartCam* presented in [2]. As seen in this figure, the PoQoCam is run under ARM-Linux 2.6.14 on the hosting processor Intel PXA255. The software-framework implements a message-based communication with the DSPs via PCI-bus.

In this work, the framework is extended by some of the presented PoQoS features. The current implementation mainly focuses in the implementation of the multi-criterion optimizer and the dynamic service manager for power-aware reconfiguration of both the DSPs and host processor. The DPM of the DSPs is based on the generic implementation presented in [7]. On the host processor, DPM is implemented by a Linux-driver that is coupled with the operating system's scheduler and implements frequency scaling and other low-power features on the PXA255. We implement the PoSeGA genetic algorithm presented in Section 4.1 as separate routine that build upon the software-framework of the *host-processor* of the PoQoCam in order to be more flexible with the dynamic power management of the DSPs.

Time	Pareto-optimal configurations found
219ms	10%
453ms	25%
1119ms	50%
1694ms	70%
2884ms	90%
7938ms	100%

Table 1. Performance evaluation of PoSeGA.

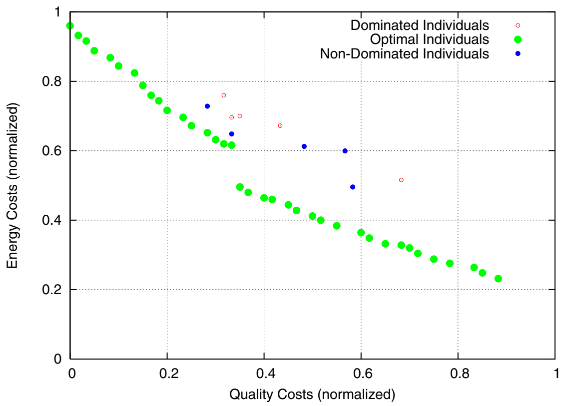
In the current implementation, the algorithm's search space (i.e., the power/service cost-model of the services and processors) is stored in an input data file that can be configured via Ethernet. The PoSeGA-algorithm itself has been implemented in ANSI-C code.

6 Experimental Results

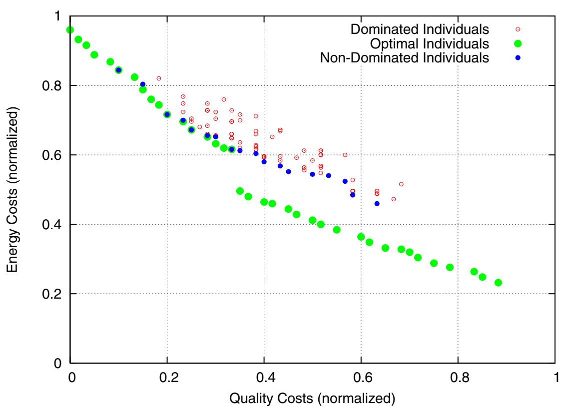
An implementation of the PoSeGA genetic algorithm has been evaluated on the PoQoCam for its use in IVS. The values for the input cost model are based on previous measurements of the DSP's processor utilization and corresponding power consumption for all different QoS-levels of the given services. In the given evaluation, PoSeGA computes the Pareto-set of camera configurations for $s = 6$ different tasks in up to $q_s = 5$ QoS-Levels and *r=2* different DSPs. The mutation- and crossover-rate has been adjusted to *m=8%* and *c=14%* following the results of previous performance evaluations.

Fig. 7 depicts three different stages of the population computed by PoSeGA as function of the costs for both optimization objectives. In order to achieve better evaluation of the obtained results, the Pareto-optimal configurations have also been computed separately by a greedy MCO-algorithm that covers the whole search space. Its results are plotted as 'optimal individuals' throughout all diagrams. Fig. 7(a) depicts the randomly generated initial population that consists of ten camera configurations, whereas already five of them are non-dominated individuals. Fig. 7(b) shows an already grown population after 30 iterations with parts of the approximated Pareto-set already converging to the real optima that represent configurations that have both DSPs activated. However, there is still a gap to the optimal individuals that represent configurations with only one DSP activated. After 240 iterations, PoSeGA generates a population that aligns with the real Pareto-set as seen in Fig. 7(c).

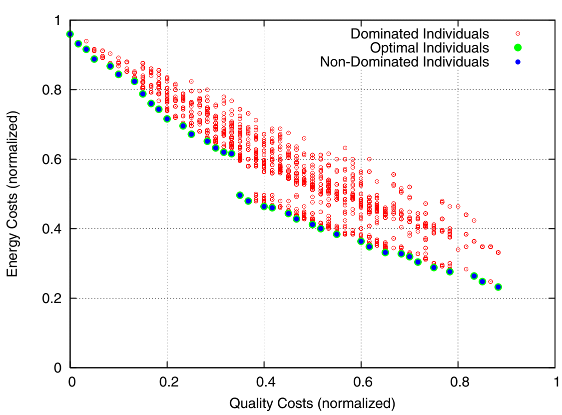
PoSeGA's performance has been evaluated on the PXA255 host processor of the PoQoCam. The results of this evaluation is listed in Table 1 for various iterative stages as percentage of real Pareto-optimal configurations that have been obtained by the greedy algorithm. As seen from these results, finding the last 10% of the Pareto-optima takes more than 60% of the overall execution time.



(a) Initial random population.



(b) Population after 30 iterations.



(c) Final population after 240 iterations.

Figure 7. Iterative stages of the MCO process.

7 Conclusion

In this paper, we presented PoQoS, a novel approach that aims in maximizing the service quality (i.e., the number of IVS-services and their QoS) while minimizing the system's power consumption. In order to find optimal camera configurations during operation, PoQoS integrates an online multi-criterion optimizer that is based on a genetic optimization algorithm, PoSeGA.

The evaluation of PoQoS on the PoQoCam, a power-efficient embedded smart camera platform, shows the feasibility of the presented approach. Future work aims the evaluation of PoQoS in a distributed experimental setup consisting of multiple embedded smart cameras.

References

[1] A. Bogliolo, L. Benini, and G. D. Micheli. A Survey of Design Techniques for System-Level Dynamic Power Management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3), Jun 2000.

[2] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach. Distributed embedded smart cameras for surveillance applications. *Computer*, 39(2):40–47, Feb. 2006.

[3] C. Chiasserini and E. Magli. Energy Consumption and Image Quality in Wireless Video-Surveillance Networks. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002.

[4] C. C. Coello. A Short Tutorial on Evolutionary Multiobjective Optimization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 21–40, 2001.

[5] R. Dick and N. Jha. COWLS: Hardware–Software Cosynthesis of Wireless Low-Power Distributed Embedded Client Server Systems. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No.1*, pages 2–16, 2004.

[6] N. Jha. Low-Power System Scheduling, Synthesis and Displays. In *IEE Proceedings of Computers and Digital Techniques, No. 20045067*, pages 344–352, 2005.

[7] A. Maier, B. Rinner, and H. Schwabach. A Hierarchical Approach for Energy-Aware Distributed Embedded Intelligent Video Surveillance. In *Proceedings of the IEEE/IFIP International Workshop on Parallel and Distributed Embedded Systems*, pages 12–16, 2005.

[8] K. Miettinen, editor. *Evolutionary Algorithms in Engineering and Computer Science*. John Wiley and Sons, 1999.

[9] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau, and N. Venkatasubramanian. Integrated Power Management for Video Streaming to Mobile Handheld Devices. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, pages 582–591, 2003.

[10] R. Steinmetz, editor. *Multimedia Systems*. Springer, Berlin, 2004.

[11] W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. *Computer*, 35(9):48–53, Sept. 2002.

Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)
1550-445X/06 $20.00 © 2006 IEEE