

Implementation and Optimization of Embedded Image Processing System

Yuwen Zheng

Shandong Women's University, Jinan, 250300, China

Keywords: embedded system, optimization process, image processing and analysis, cloud based system implementation.

Abstract. With the fast development of image science and embedded system technique, the need for the combination is bursting. In this paper, the advantages of scalability of the presented embedded vision system can be seen based on the exemplary application room monitoring. Here it's remarkable that the implementation on the low-end system is up to ten times faster than on the medium-size system. This results from the implementation of the PEG-compression in hardware. Indeed this was a lot of work but significantly reduced the hardware costs. Thus we have shown that a possible product, which is based on our vision system, can be optimized (scaled) according the required performance and unit numbers. To modify our work, selected optimization approaches and mathematical foundations are introduced to improve the performance of the system.

Introduction

Vision systems are now primarily designed and marketed for the purpose of industrial applications. Thus, the typical application fields and the demand is very different from bulk products. In connection to the cloud, a versatile new areas of application for the visual system of batch products, especially in the context of the intelligent private houses (such as security architecture and space monitoring, monitoring family of people need care, traditional reading meters of natural gas, water and electricity). All these different applications to share a common demand of preferably, size, low cost and power requirements, and the best are high communication ability. These requirements cannot be achieved product in an all-around player. As a result, the target solution is a modular scalable module combination and configuration of the embedded system. The limitation of hardware resources of embedded system is almost always the main challenge. Therefore, the main research in embedded visual focus on developing new algorithms combined with new and novel embedded hardware very specific applications such as: stereo vision: [1], robot control: [2], tracking: [3], object detection: [4], saliency map: [5] and image classification: [6]. The degree of research activity was not significant smaller focus on a particular application. Open method of the embedded vision system, but not the FPGA (field programmable gate array) which has been introduced previously. The aspects of sensor networks are treated in [7]. Systems with FPGA-based image pre-processing, but without the aspects of networking, are presented in [8]. In this paper, we propose a novel embedded image processing system, the detailed structure and in-depth analysis will be discussed in the following sections.

The Basic Principles and Concepts

The conceptual basis of our system is as followed: (1) Image pre-processing via FPGAs: FPGAs are predestinated for the implementation of simple but fast processes. Preprocessing algorithms (e.g. FIR-filter) are not very complex, but must be performed on the big amount of image data. Therefore, they can be optimally implemented on an FPGA. The specification of the algorithms is done in VHDL (Very High Speed Integrated Circuit Hardware Description Language), which allows the development of a reusable and device independent module library for fast image pre-processing. (2) Image evaluation via processors: The realization of the complex algorithm (e.g., classification) must support the processor. Here, the arm (advanced RISC machines) technology to ensure that the processing capacity scalability highest compatibility. (3) Usage of SoPC (System on a

Programmable Chip): As part of this technology, the processor and the VHDL design are implemented in a single programmable chip, significantly increasing the cost-performance ratio. (4) Conceptual anchoring of components from the consumer sector: Vision and communication components (camera, WLAN, Ethernet, zigbee) is now included in the quality of products, such as smart phones, cameras and wireless thermometer). This type of component is cheap, small and energy-efficient. (5) Use of standards: Ensure that a wide range of applications and system guarantee of the future, technology application of the standard, as much as possible. For example, the camera interface (CCIR 656), a wireless network is too slow) based on IEEE802.11 standard to provide services ARM processor technology. (6) Hierarchical software concept: Even with a consequent use of standard technologies, rescaling also always means hardware modifications, which cause repercussions on the software. Therefore, a hierarchical software concept is appreciated in which hardware-dependent layers provide hardware-independent software interfaces.

The Hardware Design and Implementation

The Overall Structure Introduction

The base of the system is made from scalable and modular hardware. The vision system can be configured from the following subsystems: Camera, ARM system and/or SoPC (Fig. 1 a b). Each of these subsystems is also scalable. In principle, three different system configurations can be identified and are presented here. In the figure, we can conclude that this version of the system consists of a camera and a Sope. The digital image data stream of a camera is preprocessed in the image preprocessing unit and the outcome (typically images) is stored in shared memory. The on-chip processor accesses the outcome via this memory and executes complex algorithms of image interpretation, configuration of image preprocessing (via shared memory), camera configuration (over I2C interface), as well as communication. The communication is performed via WLAN, USART or ZigBee.

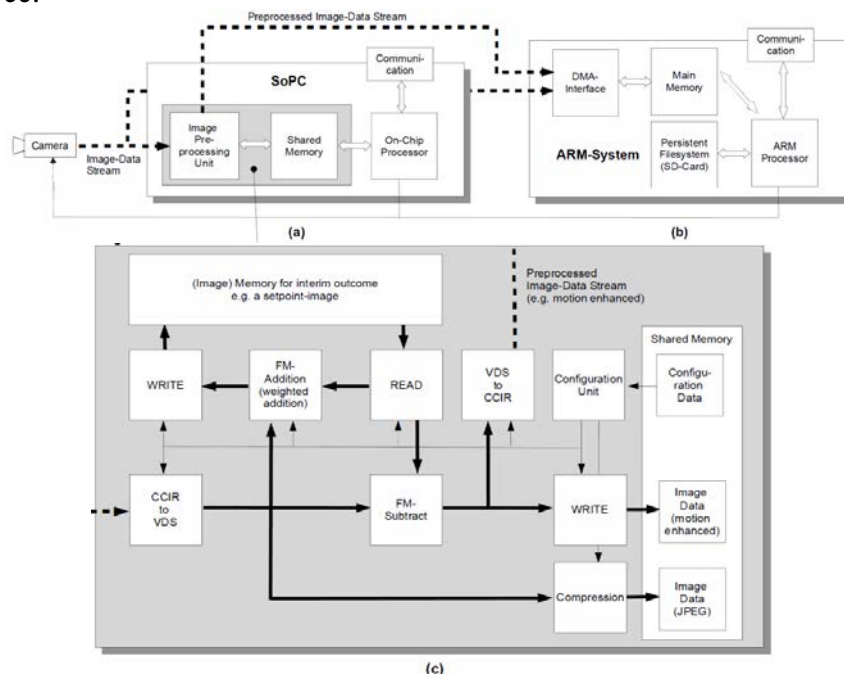


Fig.1: (a) Low-end System, (b) Medium-size System, (c) Image Pre-processing Unit

The Low-end System

This version of the system consists of a camera and a SoPC. The digital image data stream of a camera is preprocessed in the image preprocessing unit and the outcome (typically images) is stored in shared memory. The on-chip processor accesses the outcome via this memory and executes complex algorithms of image interpretation, configuration of image preprocessing (via shared memory), camera configuration (over I2C interface), as well as communication. The

communication is performed via WLAN, USART or ZigBee. The image preprocessing unit is realized on the basis of reusable VHDL modules (Fig. 1c). It operates on the image data stream (e.g. CCIR-656). Each module contains a uniform interface for a well specified internal vision data stream (VDS) and configuration data. Thus, the modules can be the modules. This parameterization is synchronous to the frame-rate clock. This avoids the changing of preprocessing parameters during one image. So-called filter modules (FM) (e.g. FM-Addition, FM-Subtract) perform signal processing on VDS. The exemplary configuration in Figure 1.c implements algorithms for motion detection and image compression. Each incoming image is subtracted from a set point-image and stored in shared memory, where it can be read by the processor and further analyzed. An appropriate dynamic configuration (setting the base address) of the WRITE module ensures that the processor operates on a double buffer. In addition, the difference image is converted into image data stream (CCIR format) and optionally can be read via DMA interface of an ARM system as a preprocessed camera image. The set point-image is dynamically adapted by a first-order IIR-Filter. This is performed by a weighted addition (FM-Addition) of the incoming image and the actual set point-image. Furthermore, each incoming image is JPEG-compressed and stored in shared memory.

The Medium-High-end System

A medium realization of the system consists of a camera and an ARM System (Fig. 1 b). The image-data stream of the camera is stored over the DMA interface directly into the main memory of the ARM processor. In addition to the image preprocessing, the ARM processor handles the image analysis, camera parameterization and communication. The communication is again performed via WLAN, USART or ZigBee. High-end systems by the camera and the SoPC arm system (Fig. 1a b). The image pre-processing indicated in SoPC. Pretreatment of image data is available: the ARM chip processor. As a result, image analysis two processor parallel execution. In addition, the ARM processor to take over the camera parameterized task, communication and data storage. Generally speaking, the arm system has lasting file system, such as the form of an SD card.

The software for the system is designed corresponding to the model of a service and implemented according the client-server concept. At the top of the software shows the visual system of service provided to the user (application). Each visual system provides a minimum vision - basic - the service (such as identification, state request, listing all of the services). In addition, the existing service image acquisition (the capture - service), each application, application services (e.g., monitoring of the room). It is essential that the application service provided by the vision system can be used (as client-process) another vision system services (such as remote-file-service). This ensures transparency about service deployment "location", thus simple visual system scalability by building a network. The vision-communication-lib can be understood as a "virtual library", under which all client functions of all services are subsumed. In practice, depending on various application areas of the vision system, meaningful subsets are defined and bound in real libraries. The client functions are implemented according to the concept of RPCs (Remote Procedure Calls). Depending on the used underlying communication interface (Sensor Cloud-API or packet-communication-lib) the service of the vision system can be reached via Cloud or local network. The vision-algorithm-lib can also be understood as a virtual library, under which all standard functions for image processing and image analysis are summarized. It is important that all these algorithm functions use a uniform access to the image data, like provided by the IBV-Lib (our library for industrial image processing). The vision-file-lib provides a uniform interface for file access. It ensures a transparency regarding the location (local, remote) of the file and the type of the storage (persistent e.g. on SD-Card, non-persistent e.g. on RAM-Disc). The packet communication- lib provides a uniform packet-oriented interface for the communication via different communication mediums (LAN, WLAN, USART, ZigBee), between the different system platforms (SoPC, ARM system using Linux, Android, Windows). The capture-lib and the camera-config-lib provide a uniform platform-independent interface to the camera. The lowest software layer is platform-dependent and mostly self-explanatory, based on the previously given context.

The Mathematical Based Optimization

The Goal of Optimization

An optimization problem can usually be posed in various, mathematically equivalent, ways. These mathematical formulations differ in the type of constraints and optimization variables. Different formulations are obtained by using parts of the constraints to eliminate variables, or by exposing variables to the optimization and introducing coupling constraints. Such problem formulations can have a great impact on how the optimization problem can be solved efficiently. The coupling of variables and the structure corresponding to a given problem formulation can make some formulations more suitable for certain optimization methods. The formulations together with the choice of particular solution algorithms also determine to which extent, and of which content, problem information has to be provided to the optimization algorithm. Problem formulation issues are addressed in various places for particular problems and in the general optimization literature. We present the formulations for the real, possibly nonlinear, equality-constrained optimization problem:

$$\min_{(\delta, u)} f(\delta, u) \quad \text{s.t. } c(\delta, u) = 0 \quad (1)$$

Obviously, the number and type of variables and constraints change with the problem. Measured in number of variables and constraints, the all-at-once formulation is typically by far the largest problem. The black-box formulation typically leads to the smallest problem. But the expense of a single function evaluation is typically significantly higher for the black-box formulation than for the all-at-once formulation. This is due to the fact that the objective function in the black-box case requires the evaluation of objective function. This evaluation often corresponds to the solution of nonlinear systems. Such problem formulations can have a great impact on how the optimization problem can be solved efficiently. The coupling of variables and the structure corresponding to a given problem formulation can make some formulations more suitable for certain optimization methods. The increase in cost of function evaluations when switching from the all-at-once to the black-box case also carries over to the derivatives. Availability and, in second line, ease of derivative computations can have considerable impact on the choice of problem formulation. So can the size of the quantities involved.

The Reverse Hilbert-type Inequalities

In the literature review, we adopt the basic concept of reverse inequality to modify our model [8-10]. By introducing two pairs of conjugate exponents and estimating the weight coefficients, we establish reverse versions of Hilbert-type inequalities, as described by Jin (J. Math. Anal. Appl. 340:932-942, 2008), and we prove that the constant factors are the best possible. As applications, some particular results are considered. Under the same conditions, there are some Hilbert-type inequalities that are similar to, which also have been studied and generalized by some mathematicians. Recently, by studying a Hilbert-type operator, Jin obtained a new bilinear operator inequality with the norm, and he provided some new Hilbert-type inequalities with the best constant factor. We define the formula in the equation 2-4:

$$\sum_{n=1}^{\infty} \sum_{m=1}^{\infty} k(m, n) a_m b_n < k_r \left\{ \sum_{n=1}^{\infty} n^{p/r-1} a_n \right\} \quad (2)$$

$$k_r \left\{ \sum_{n=1}^{\infty} n^{p/r-1} a_n \right\} < (k_r)^p \sum_{n=1}^{\infty} n^{p/r-1} a_n^p$$

$$\sum_{n=1}^{\infty} \sum_{m=1}^{\infty} g(m, n)^p a_m b_n < g_r \left\{ \sum_{n=1}^{\infty} n^{p/r-1} a_n \right\}^p \quad (3)$$

$$g_r^p \left\{ \sum_{n=1}^{\infty} n^{p/r-1} a_n \right\}^q < \{(g_r)^p \sum_{n=1}^{\infty} n^{p/r-1} a_n^p\}^q$$

$$\sum_{n=1}^{\infty} \sum_{m=1}^{\infty} g(m, n)^p a_m b_n < g_r \iint n^{p/r-1} a_n^p \left\{ \sum_{n=1}^{\infty} n^{p/r-1} a_n \right\}^q \quad (4)$$

Through formula 2-4, we could conduct the optimization work.

The Experimental Analysis

The Set-up of Experiment

As shown in Figure 2, the vision system is part of a cloud wide network (Sensor Cloud), which extends over a large number of facilities (private homes, industry buildings etc). Each facility has its own local network. Typically all communication between this local network and the cloud is passing through an access point (location master), which separates the secure cloud communication world from the less secure internal world of a home. Further on it connects several house automation buses to allow interworking of sensors and actors using different standards (e.g. KNX and Homematic sensors) based on sensor ontologies. In respect to highest reliability we decided to foresee measurements for unreachability times. Therefore the buffering in a local database is one more task of the location master.

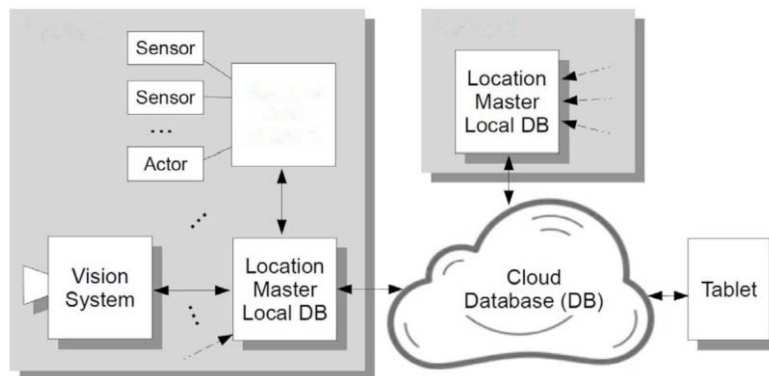


Fig.2: The Sensor Cloud

The Result Discussion

The monitoring is based on the comparison of a cyclically recorded actual image with a dynamic time adapted set-point image. The two images are divided into equal-sized tiles, and tiles at the same position are compared. An alarm is raised if a sufficient number of defined tiles display differences high enough, for an adequately long duration, between the actual image and the set-point image. An interactive parameterization of the system can be done via cloud, using an Android-App as the client for the application-service "room-monitoring". This app also receives and visualizes the alarms. Upon receipt of an alarm, the user can request relevant images from the scene and thus pursue the incident online. A second implementation is based on a low-end system configuration (Zynq-7020, WLAN-module, OV96S0-cameramodule). For this configuration, we additionally designed and implemented a VHDL-module for hardware-based IPEG compression as part of the image preprocessing unit. Now the image compression can be done in real-time (O.OSs). The related results to our hardware-based IPEG compression would overload the scope of this paper and will be presented in further publication. The cloud database system is designed as a federated database (FDBS). A global schema describes the cloud components and the local databases of the FDBS and all their entities. It contains all properties (relationships, namespaces, data types and further important information) of the cloud database. One version of the global schema is written in ISON format. The compact format of ISON allows a simple structure to represent the database entities.

Conclusions

The advantages of scalability of the presented embedded vision system can be seen based on the exemplary application room monitoring. Here it's remarkable that the implementation on the low-end system is up to ten times faster than on the medium-size system. This results from the implementation of the IPEG-compression in hardware. Indeed this was a lot of work but significantly reduced the hardware costs. Thus we have shown that a possible product, which is based on our vision system, can be optimized (scaled) according the required performance and unit

numbers.

Acknowledgements

The research work was supported by Shandong Provincial Staff Education office No. 2013-324.

References

- [1] Chon, Kye-Sung. "Traveler destination image modification process and its marketing implications." Proceedings of the 1990 Academy of Marketing Science (AMS) Annual Conference. Springer International Publishing, 2015.
- [2] Hays, James, and Alexei A. Efros. "Large-Scale Image Geolocalization." Multimodal Location Estimation of Videos and Images. Springer International Publishing, 2015. 41-62.
- [3] Ponomarenko, Nikolay, et al. "Image database TID2013: Peculiarities, results and perspectives." Signal Processing: Image Communication 30 (2015): 57-77.
- [4] Bernatowicz, K., et al. "Quantifying the impact of respiratory-gated 4D CT acquisition on thoracic image quality: A digital phantom study." Medical physics 42.1 (2015): 324-334.
- [5] Herrell, S. Duke, Robert L. Galloway Jr, and Michael I. Miga. "Image Guidance in Robotic-Assisted Renal Surgery." Advances in Image-Guided Urologic Surgery. Springer New York, 2015. 221-241.
- [6] Shih, I. Hong, David Yang, and Fanlin Kong. "Image guidance of hexosamine pathway-directed therapy in oncology." Society of Nuclear Medicine Annual Meeting Abstracts. Vol. 55. No. Supplement 1. 2014.
- [7] Hodge, Francis. Yankee Theatre: The Image of America on the Stage, 1825–1850. University of Texas Press, 2014.
- [8] Shulimzon, Tiberiu R. "Reply: Is It Actually a pCLE Image of a Sarcoid Granuloma" American journal of respiratory and critical care medicine 189.2 (2014): 236-236.
- [9] Xu, Biao, et al. "On reverse Hilbert-type inequalities." Journal of Inequalities and Applications 2014.1 (2014): 198.
- [10] Yang, Bicheng, and Qiang Chen. "A more accurate half-discrete reverse Hilbert-type inequality with a non-homogeneous kernel." Journal of Inequalities and Applications 2014.1 (2014): 1-13.