Smart Cameras: A Review (2)

Yu Shi and Serge Lichman
Interfaces, Machines And Graphical ENvironments (IMAGEN)
National Information and Communications Technology Australia (NICTA)
yu.shi@nicta.com.au, serge.lichman@nicta.com.au

This is the second and last part about about Smart Cameras provided by Yu Shi and Serge Lichman

4 Review of ASIP Algorithms for Smart Cameras and State-of-the-Art Systems

If cameras are extensions of human eyes, the smart cameras are pushing the boundary of possibilities to become extensions of human brain as well. What makes a camera smart is the intelligent and application specific information processing (ASIP) algorithms that are built into the software architecture of the camera systems. In this section we firstly explore some common characteristics of intelligent algorithms for smart cameras. We then review several categories of algorithms as applied to machine vision, surveillance and other prominent applications, and some state-of-the-art smart camera systems in use in these applications areas.

4.1 Common Characteristics of Algorithms for Smart Cameras

The primary function of a smart camera is to conduct autonomous analysis of the content of an image or video and achieve a high-level understanding of what is happening in the scene. One of the most commonly adopted approaches is image processing-based pattern recognition, which is a branch of artificial intelligence. Pattern recognition assumes that the image may contain one or more objects and that each object belongs to one of several predetermined types or classes. Given a digitized image containing several objects, the pattern recognition process consists of three main phases, each including several processing tasks:

- Signal level processing – image enhancement, image segmentation;
- Feature level processing – feature extraction, feature measurements and tracking; and
- Object level processing – object classification and estimation.

This is illustrated in figure 5. Also shown in figure 5 is a semantic-level processing component, which is central to the output or action side of smart cameras. The main tasks at this level include possible joint analysis of inputs from additional cameras, other sensory and database inputs, data fusion, event description, control signal generation. It should be noted that some tasks at different levels or phases may intersect each other during processing.

Image segmentation at signal level is essential to all subsequent processing tasks, aiming at dividing an image into distinct parts, each having a common characteristic. Image segmentation can be based on color, texture, shape and motion. Feature extraction is crucial to pattern recognition. This is where the segmented regions or objects are measured. A measurement is the value of some quantitative property of an object. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristic of the object. This drastically reduced amount of information (compared to the original image) represents all the knowledge upon which the subsequent classification decision must be based. Object classification outputs a decision regarding the class to which each object

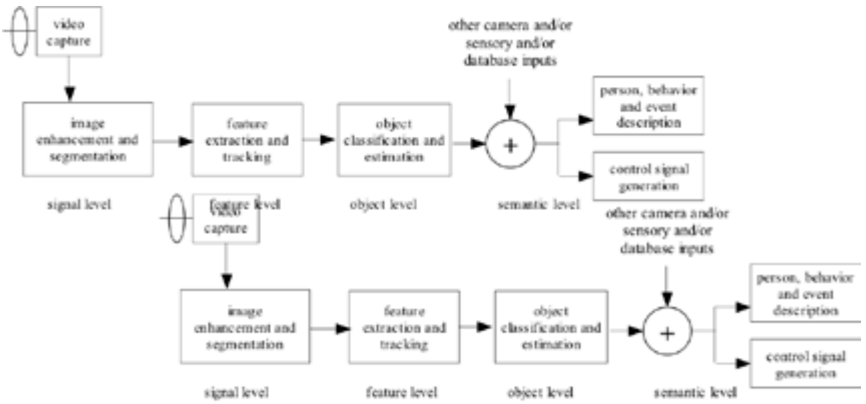


Figure 5: General processing flow of algorithms for pattern recognition and smart cameras.

belongs. Each object is recognized as being of one particular type, and the recognition is implemented as a classification process.

For simple applications, not all these levels and tasks are required to be implemented. For example, the camera in an optical mouse only performs signal- and feature-level processing tasks. On the other hand, for a particular processing task, different applications can have quite different requirements on the camera's performance, robustness and reliability. For example, the requirements for robustness of processing tasks at all levels are much higher for video surveillance monitoring human movement and behaviors than for industry machine vision cameras performing parts inspection or sorting.

Tasks at signal- and feature-levels are usually data-intensive and are well suited for hardware-based implementation to meet speed demands. Tasks at the object level can be math-intensive and may need high performance processor(s) to complete. Stand-alone smart cameras built on a multi-processor architecture would have one processor, such as a DSP or an FPGA, to perform tasks at signal- and feature-levels, and have a high performance DSP or RISC microprocessor to perform statistical object classification.

When designing smart cameras as embedded systems for demanding applications such as surveillance and automobiles, there are several important and challenging issues that need to be addressed, such as the development of low-complexity, low-cost algorithms suitable for hardware implementation, and software and hardware co-design, in order to map algorithmic requirements to hardware resources. These issues will be further discussed in section 5.1.

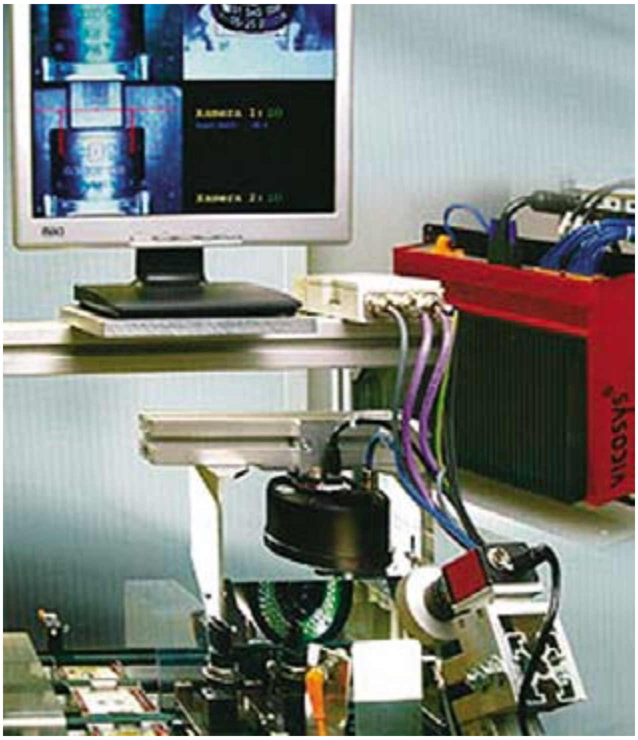
4.2 Application: Intelligent Video Surveillance Systems (IVSS)

4.2.1 Current Research in Algorithms for IVSS

Video surveillance in dynamic scenes, especially for humans and vehicles, is currently one of the most active research topics in computer vision and pattern recognition. The IEEE and IEE have organized many workshops and conferences on intelligent visual surveillance in the last several years and have published special journal issues that focus solely on visual surveillance or in human motion/behavior analysis. Hu and Valera recently conducted excellent surveys on various algorithms and techniques under research and development for video surveillance. They also reviewed some high profile IVSS systems. Some comments in this section are derived from their papers.

For video surveillance, image segmentation most often starts with motion detection, which aims at segmenting regions corresponding to moving objects from the rest of an image. Background modeling is indispensable to motion detection. 3-D models can provide more realistic background descriptions but are more costly. 2-D models have more applications currently due to their simplicity. However, all modeling techniques need to find ways to reduce the effect of unfavorable factors such as illumination variation, moving shadows and so on. Promising techniques for motion segmentation include simple background subtraction, temporal differencing, and more complex optical flow methods. Skin-color based segmentation can be very useful when human objects are close enough to the camera and lighting is consistent. Once segmentation has provided isolated objects, feature extraction and measurements can be performed on each object. Simple algorithms for feature extraction include image moments, which can provide geometrical features of the objects. For gesture and behavior recognition, promising algorithms for feature extraction include MEF (Most Expressive Features), extracted by Karhunen-Loeve projection, and MDF (Most Discriminative Features), extracted by multivariate discriminate analysis. Since sometimes it is not easy to specify features explicitly, in some applications when the image size is small enough, the whole image or transformed image is taken as the feature vector. Examples of algorithms for object classification are shape-based classification and motion-based classification. After motion detection and object classification, video surveillance systems generally track moving objects from one frame to another. Promising algorithms for object tracking can be classified into four categories: region-based tracking, active contour based tracking, feature based tracking, and model based tracking. Particle filters have recently become a major way of tracking moving objects.

Human behavior understanding and personal identification are among the most challenging tasks facing IVSS systems for high-end security applications. Behavior understanding involves the analysis and recognition of motion patterns, and the production



of high-level description of actions and interactions. Promising approaches and algorithms for behavior understanding include dynamic time warping, finite state-machine, HMMs (Hidden Markov Models), time-delay neural networks. Personal identification is of increasing importance for many security applications. The human face and gait are now regarded as the main biometric features that can be used for personal identification in video surveillance systems. While face recognition research and development has made a lot of progress in recent years, current research on gait recognition is still in its infancy.

4.2.2 State-of-the-Art IVSSes

A number of high-profile IVSSes have been reported in recent years. These systems, some deployed in real-world applications, applied various pattern recognition techniques described in previous sections and provided features such as people tracking, behavior recognition, detection of unattended objects and so on. Examples are the real-time visual surveillance system W4, the Pfinder system developed by Wren, the single-person tracking system, TI, developed by Olsen, and a system at CMU (Carnegie Mellon University) that can monitor activities over a large area using multiple cameras connected by a network.

A few IVSSes based on the use of stand-alone smart cameras have also been reported. The V2 system developed by Christensen and Alblas is a surveillance system that avoids the disadvantages of the centralized computer server, and moves many of the processing tasks directly to the camera, making the system a group of smart cameras connected across the network. The event detection and storage of event video can be performed autonomously by the camera. Thus, normally, it is only necessary to communicate with a central point when significant events occur. The VSAM project described by Collins is a multi-camera surveillance system composed of a network of ‘smart’ sensors that are independent and autonomous vision modules. These vision sensors are capable of detecting and tracking objects, classifying the moving objects into semantic categories such as ‘human’ or ‘vehicle’ and identifying simple human movements such as walking. Desurmont developed a smart network camera system with three smart cameras to perform people tracking and counting in shopping malls. Their system uses web services standards and XML-based meta data to implement inter-camera and camera-to-host coordination. Fleck designed a smart camera that contains an FPGA and a PowerPC processor to perform face tracking and people tracking, using particle filters on HSV (Hue, Saturation, Value) color distributions. The camera outputs the approximated PDF (probability distribution function) of the target state to a host computer.

4.3 Application: Industry Machine Vision

While advanced algorithms for smart cameras for surveillance applications are mostly still in their research and development stage, due to high complexity and high-level of robustness requirement for real-world

applications, smart cameras for industry machine vision have long established their places in the market as mature players. Most machine vision cameras are stand-alone and autonomous smart cameras, where communications with PC or other central control unit is only needed for camera configuration, firmware upgrading or in some cases output data collection. Most algorithms implemented in these cameras follow the similar processing flow described in figure 5. One important reason for the relative maturity of machine vision smart cameras, compared with smart cameras for surveillance, is that the application requirements for machine vision cameras are much less restrictive compared with those for surveillance cameras. In other words, many pattern recognition algorithms or techniques have a much better chance of performing with satisfactory robustness and reliability for machine vision than for surveillance applications. This is because machine vision cameras mainly deal with conditions such as:

- indoor use, thus good and consistent lighting conditions can be more easily guaranteed;
- minimum problems of occlusion;
- static and known background, thus unusual feature detection is simpler;
- limited object patterns to be recognized; and
- no human movement tracking and recognition is necessary.

There are many proven software packages on the market that can be customized or directly implemented for programmable machine vision cameras. Most of these packages are for special industry sectors, but some are general purpose packages, including a few powerful up-market libraries such as Halcon library. The Halcon library provides algorithms that include shape-based matching to find objects based on ROI (region of interest) modeling, blob analysis, metrology (both 1D and 3D), edge detection, edge and line extraction, contour processing, template matching, and color processing.

Thanks to the advancements in embedded system technologies and improved affordability of processing power, there is a migration of the functionality of what were once only PC-based systems down to the smart camera level. Artificial intelligence is one of these functionalities. Pulnix America's ZiCAM camera, for example, makes use of a hardware neural network to eliminate the need for programming to execute image-understanding algorithms. It can learn what is required for a machine vision application, and once taught, operates as a stand-alone smart camera. Wintriss Engineering manufactured a smart camera which sports a microprocessor, DSP and multiple FPGAs with up to 130,000 gates. The company offers both area- and line-scan versions of their smart cameras, with line scan version being able to perform imaging-related processes on 5 150 pixel lines at 40 MHz. One such camera uses an FPGA to perform image sensor control and pixel correction, and the combination of the compute power in the camera head to run real-time digital filters, lighting correction, streak correction and input/output capability. Ultimately geometric and photometric manifested flaws are discriminated based on connectivity analysis, all performed within the camera.

4.4 Application: Intelligent Transport Systems and Automobiles

4.4.1 ITS Applications

There is growing awareness and interest in using smart cameras in Intelligent Transport Systems (ITS) and automobile industries. IEEE organized very recently an international workshop in June 2005 on Machine Vision for Intelligent Vehicles. Generally speaking, the application and algorithmic requirements for ITS are quite similar to those of IVSS. These requirements can be quite different for automobile applications, however, where high-speed imaging and processing are often needed, imposing higher level of demand on both hardware and software. Increased robustness is also required for car-mounted cameras to deal with varying weather conditions, speeds, road conditions, car vibrations. CMOS image sensors can overcome problems like large intensity contrasts due to weather conditions or road lights and further blooming, which is an inherent weakness of existing CCD image sensors.

There have been a number of successful applications of smart camera systems for ITS reported in the literature. The VIEWS system at the University of Reading is a 3D model-based vehicle tracking system. Kumar described a real-time rule-based behavior-recognition system for traffic videos. This system will be useful for better traffic rule enforcement by detecting and signaling improper behaviors, which is capable of detecting potential accident situations and is designed for existing camera setups on road networks. Beymer presented a smart camera-based monitoring system for measuring traffic parameters. The aim of the system is to capture video from cameras that are placed on poles or other structures looking down at traffic. Once the video is captured, digitized and processed by onsite smart camera, it is transmitted in summary form to a transportation management centre for computing multi-site statistics like travel times.

Bramberger described an embedded smart camera for stationary vehicle detection. They discussed the mapping of high-level algorithms to embedded system components. Dimitropoulos described a network of smart cameras deployed at the airport to detect and track aircrafts; each camera can autonomously detect aircraft traffic in multiple locations within its field of view. A camera data fusion module performs data fusion from multiple cameras to determine the location and size of the aircraft. Other applications for smart cameras for ITS include vehicle behavior in parking lots, vision based vehicle speed measurement, red-light intrusion at traffic lights, vehicle number plate recognition. Some authors have expressed the need to integrate smart traffic surveillance systems with existing traffic control systems to develop the next generation of advanced traffic control and management system.

4.4.2 Automobile Applications

Intelligent vehicles will form an integral aspect of the next generation technology of ITS. Smart camera-powered intelligent vehicles will have the comprehensive capability of monitoring the vehicle environment including the driver's state and attention inside of the vehicle as well as detecting roads and obstacles outside the vehicle, so as to provide assistance to drivers and avoid accidents in emergencies. However, building and integrating smart cameras into vehicles is not an easy task: On one hand the algorithms require considerable computing power to work reliably in real-time and under a wide range of lighting conditions. On the other hand, the cost must be kept low, the package size must be small and the power consumption must be low. Applications of smart cameras in intelligent vehicles include lane departure detection, cruise control, parking assistance, blind-spot warning, driver fatigue detection, occupant classification and identification, obstacle and pedestrian detection, intersection-collision warning, overtaking vehicle detection. Below are a few examples.

Stein described a single smart camera-based adaptive cruise control system for intelligent vehicles. In a paper

on obstacle detection using stereo vision, Ruichek focused on a multilevel- and neural-network-based stereo-matching method for real-time road obstacle detection with linear cameras for use in vehicles. Xu addressed the problem of pedestrian detection and tracking with night vision using a single infrared video camera installed on the vehicle. The EyeQ is a single chip smart camera processor developed by Mobileye. It has been fabricated using 0.18μm CMOS technology, operating at 120 MHz. It integrates two 32 bit RISC ARM946E CPUs, four Vision Computing Engines, a multi-channel DMA (Direct Memory Access) and several peripherals and is designed for computationally intensive applications for real-time visual recognition and scene interpretation for use in intelligent vehicle systems.

4.5 Other Application Areas

Other important applications for smart cameras include HCI, medical imaging, robotics, games and toys. Optical mice are widely used. Smart cameras performing gesture recognition will play important role in the development of multimodal user interfaces. Bonato presented an FPGA-based smart vision system for mobile robots capable of performing real-time human gesture recognition. The RVT system developed by Leeser and based on FPGA processing allows surgeons to see live retinal images with vasculature highlighted in real time during surgery.

5 Smart Camera Design Considerations and Future Directions

In this final section we discuss design considerations for smart cameras as embedded systems, identify several key issues that need to be addressed by the design and research community, and speculate on the future directions of smart camera research and development.

5.1 Design Considerations

5.1.1 Design and Development Process

Figure 6 shows a typical design and development process for smart cameras as embedded systems (excluding single-chip smart cameras). A shown in figure 6, the process can be iterative, especially if the initial application specification was not complete from the end user point of view.

The system architecture design stage will decide on software and hardware architectures, based on performance, deadline and cost criteria. Algorithmic design and timing design suitable to the targeted hardware platform also needs to be defined. The mapping between algorithm requirements and hardware resources is an important issue. The proof-of-concept stage may use a PC platform for research and algorithm development. Usually a COTS (Commercial Off-The-Shelf) general purpose camera is used at this stage. Hardware components need to be acquired, integrated and tested. However, this is not needed if, during the architecture design stage, a third party camera development platform or hardware accelerator unit for video processing is identified to be an appropriate solution to hardware platform (see section 5.1.6 for examples of smart camera development platforms). The algorithm conversion stage includes tasks such as converting floating-point arithmetic to fixed-point arithmetic, low power and low complexity version consideration, implementation using HDL (Hardware Description Language). The Embedded System Integration stage will result in a prototype smart camera using an embedded hardware platform running embedded versions of algorithms.

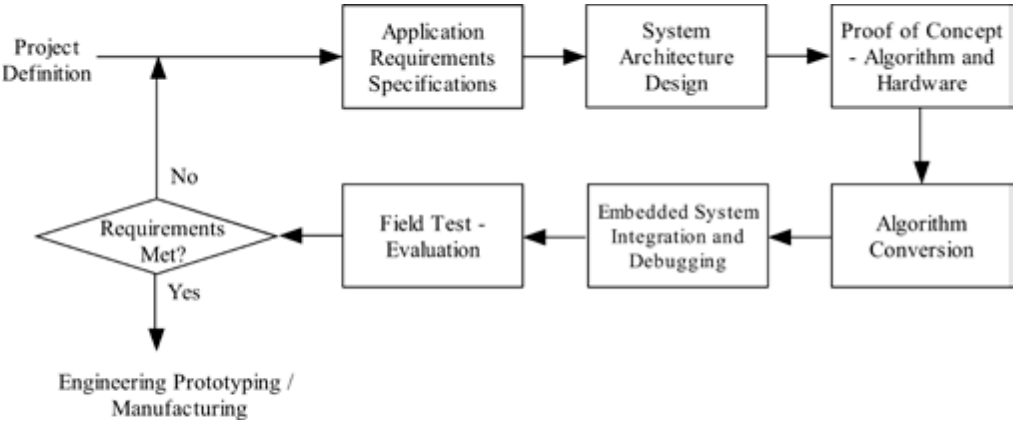


Figure 6: Design and development process for smart cameras as embedded sys-

5.1.2 System Architecture and Design Methodology

System architecture design will surely depend on application requirements, which can be very simple (e.g. an optical mouse) but can be very complex (e.g. face recognition). System architecture design has to consider many factors such as the hardware platform, cost, time to market, flexibility, and so on. Generally speaking, a heterogeneous, multiple-processor architecture can be ideal for smart camera development. For example, such an architecture may consist of an FPGA or a DSP as a data processor to tackle image segmentation and feature extraction, and a high-performance DSP or media processor to tackle math-intensive tasks such as statistical pattern classification. This kind of system can allow better exploitation of pipelining and parallel processing, which are essential to achieve high frame rates and low latency. Some authors have reported work on the impact of hardware system architecture on the level of implementable pipelining and parallel processing for smart cameras. Some initial work has been reported on design methodology for embedded vision systems.

5.1.3 Embedded Processors

There are generally four main families of embedded processors that can be used for smart cameras: Microcontrollers, ASICs (Application Specific Integration Circuits), DSPs (Digital Signal Processors) and PLDs (Programmable Logic Devices) such as the FPGA. Microcontrollers are cheap but have limited processing power and are generally not suited for building demanding smart cameras. ASICs are powerful and power-efficient processors, but the design cost and risk are high and they are viable solutions only when volume is high and time-to-market is well-timed. DSPs are relatively cheap and powerful in performing image and video processing, but for demanding applications usually more than one DSP would be needed. DSP-based solutions can be cost-effective for medium-volume production. Recently a new class of DSP processors, called media processors, has come into the vision market. Media processors try to provide a good trade-off between flexibility and cost-effectiveness. They typically have a high-end DSP core employing SIMD (Single Instruction Multiple Data) and VLSI architectures, married on-chip with some typical multimedia peripherals such as video ports, networking support, and other fast data ports. Examples of media processors are Philip's TriMedia, TI's DM64x, ADI's (Analog Devices, Inc) Blackfin.

The FPGA has recently emerged as a very good hardware platform candidate for embedded vision systems such as smart cameras. One of the most important advantages of the FPGA is the ability to exploit the inherently parallel nature of many vision algorithms. FPGAs used to be mainly employed as glue logic between processors and peripherals, but the introduction of on-chip hardware multipliers and dual-port memory has made FPGAs excellent options for DSP applications. The integration of microprocessors into FPGA chips (such as Xilinx' Virtex-II Pro and Virtex-4 chips) made them true system-on-a-chip solutions. These features, together with the continuous improvements in cost and maturity of design tools, have made FPGAs very competitive against DSPs and media processors for many types of embedded vision system designs. In fact, an increasing number of publications on smart cameras as embedded systems have employed FPGAs as the sole processor or as a data-intensive processor before a DSP or a media processor, in a powerful heterogeneous multi-processor architecture. Sen has recently proposed a design methodology for effectively and efficiently implementing computer vision algorithms on FPGA to build smart cameras. A study to compare the relative performance of running various image processing routines on DSP, PowerPC, Intel Pentium 4 and FPGA was published on Alacron's web site, in which the FPGA solution was found to produce a distinct advantage. However, a more standardized performance evaluation mechanism to help processor selection is much needed.

How should one choose between DSPs, media processors, ASICs and FPGAs? Kisacanin proposed a practical way to help processor selection based on intended production volume, cost and development flexibility. He argued that ASICs may be suitable for high volume of over 1 000 000units, DSPs or media processors for medium volumes between 10 000 and 100 000 units, while for low volumes of under 10 000, FPGAs can be a good viable candidate.

5.1.4 Algorithms Development and Conversion

Algorithm development for embedded systems is quite different from that for PC-based platforms. Basically it can be a lot more demanding and challenging, especially if FPGA or ASIC processors are targeted. Usually when designing applications for ASIC or FPGA, one has to understand chip architecture so that algorithms can be executed efficiently and effectively. Nowadays behavior synthesizers or algorithmic synthesizers do exist to help designers to forget about the device architecture and focus on functionality, but they come at the cost of efficiency in terms of chip area or gate counts and power consumption. Therefore, it is always important to gain an intimate knowledge of the device architecture of whichever of the ASIC, FPGA or DSP is targeted. This intimate knowledge can also help design parallel processing and pipelining processing, which can be a very important and effective video processing technique. Converting floating-point arithmetics to fixed-point and eliminating divisions as much as possible (by using hardware multipliers and look-up tables, for example) are other design considerations for algorithm conversion.

5.1.5 Other Factors

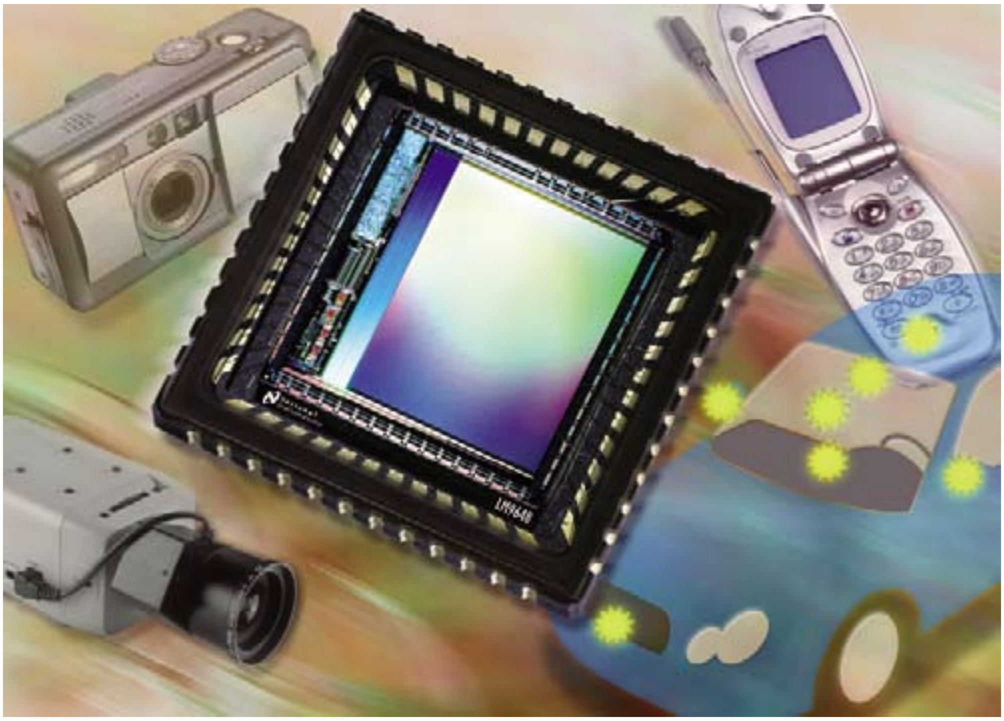
Memory System - Smart cameras need flexible memory models to meet requirements such as scalable frame buffers to cope with increasing image sensor resolutions. As the smart camera may integrate different types of processors, the memory system should support potentially complex processing pipeline and parallelism in order to meet the application's real-time requirements. For single chip smart cameras, care needs to be taken at design stage to conserve memory.

Communication Protocols - There are currently too many data output protocols for cameras, such as Firewire, CameraLink, GigE, USB. Firewire is maturing but CameraLink remains the bandwidth leader and very popular with the machine vision users. Unfortunately, the variety of digital interfaces increases the confusion in the market and put pressure on the camera vendors to support multiple versions of cameras with different interfaces.

5.1.6 Smart Camera Development Platforms

There have been a number of commercially available programmable smart camera platforms for developers to design and prototype smart cameras for applications such as machine vision, biometrics, HCI and surveillance. Philips has introduced the INCA (INtelligent CAmera) series of programmable cameras which integrate CMOS image sensors of various resolutions and a highly flexible duel-core processing unit which includes a Xetal processor for computation intensive signal processing such as feature extraction, together with a high performance TriMedia DSP core for math-intensive processing tasks such as pattern recognition. The camera comes with an application development kit allowing for fast prototyping. One application has been designed for face recognition, in which the Xetal is used for face detection and TriMedia for face recognition. Sony has recently released a smart camera development system XCI-SX1 that integrates an SXGA CCD image sensor (15 frames per second, 34fps at 640x480 resolution) and an AMD GeodeGX533 400Mhz processor running MontaVista Linux operating system. The camera platform is designed to provide OEMs, systems integrators and vision tool manufacturers a rugged, robust component, combining the imager, intelligence and interface in a single plug-in module that is simple to set up and easy to integrate. The IQeye3 IP camera from IQinvision Inc, powered by a 250 MIPS PowerPC CPU, is a platform for smart IP network camera development.

Some signal processing tool development companies provide multi-processor development systems that can serve as excellent development platforms for smart cameras. For example, Hunt Engineering provides a development platform HERON based on a Xilinx FPGA and a TI (Texas Instruments) DSP. They also provide expansion capabilities to integrate video capture, IPs, more DSPs and/or FPGAs for creating scalable smart camera architectures. Lyrtech also provides similar development systems in its SignalMaster series of products. These systems generally provide flexible communication ports and drivers.



5.2 Key Issues or Challenges

System Design – The proprietary nature of smart cameras can limit choices of hardware, like imagers, I/O, lighting, lens and the communications format. This may lead to a lack of expandability and flexibility of PC-based systems. On the other hand, smart cameras don't have as many software applications and libraries as already exist for PC/frame grabber-based systems. In terms of design methodology, the easy integration of intellectual property in the design tool and flow can help foster product differentiation. Other important system-level issues include smart camera operating systems, development tools.

CMOS Image Sensors – Dynamic range is still one of the key aspects where CMOS image sensors lag behind CCD. Improvement in this area can lead to more low-cost smart cameras using CMOS image sensors for machine vision and surveillance applications.

Algorithm Development – Many intelligent pattern recognition algorithms work well in laboratory conditions but fail when deployed and implemented in real-world conditions (occlusion, lighting condition changes, unfavourable weather conditions), and embedded system environments (scant resources, low power, low cost). Robustness and low complexity are among key issues facing researchers developing algorithms for smart cameras in surveillance, ITS and automobile applications.

Performance Evaluation - This is a very significant challenge in smart surveillance systems. Evaluating the performance of video analysis systems requires significant amounts of annotated data. Typically, annotation is a very expensive and tedious process. Additionally, there can be significant errors in annotation. All of these issues make performance evaluation a significant challenge.

Standards Development – There is need for the development of some smart camera standards. In fact, the European Machine Vision Association (EMVA) has recently launched an initiative (EMVA 1288 Standard) to define a unified method to measure, compute and present specification parameters for smart cameras and image sensors used for machine vision applications. More needs to be done in this respect.

Single Chip Smart Cameras – Single-chip smart cameras are an attractive concept, but the manufacturing cost for the single-chip smart cameras can be high because the feature size for making digital processors and memory is often different from the one used to make image sensors, which may require relatively large pixels to efficiently collect light. Therefore, for applications where physical space and power consumption is not extremely restrictive, it probably still makes sense to design the smart camera in a multi-chip approach with a separate image sensor chip. Separating the sensor and the processor also makes sense at the architectural level, given the well-understood and simple interface between the sensor and the computation engine.

5.3 Future Directions

The demand for smart cameras will steadily increase in traditional industries such as surveillance and industry machine vision, and may also come from new industry and market segments such as healthcare, entertainment, education and so on. Research interest, economic and social factors will drive continuous technological and product development. Based on the discussions above, we can discern the following future directions for smart camera system and technologies.

• At the system design level, continuous effort will be made in the development of a research strategy or design methodology for smart cameras as embedded systems. Same for the development of libraries and tools that facilitate algorithm implementation in DSPs and FPGAs. Research on the general and 'optimal" architectures for smart cameras and on real-time operating systems for smart cameras will be undertaken, and the issue of too many digital interfaces (Firewire, CameraLink, etc) for cameras will be addressed.

• At the ASIP algorithm development level, in order to improve performance and robustness of existing techniques, research should address issues such as occlusion handling, fusion of 2D and 3D tracking, anomaly detection and behavior prediction, combination of video surveillance and biometrical personal identification, multi-sensory data fusion.

• Multi-modal, multi-sensory augmented video surveillance systems have the potential to provide improved performance and robustness. Such systems should be adaptable enough to adjust automatically and cope with changes in the environment like lighting, scene geometry or scene activity.

• Work on distributed (or networked) IVSS should not be limited to the territory of computer vision laboratories, but should involve telecommunication companies and network service providers, and should take into account system engineering issues.

• In the machine vision arena, smart cameras will offer more and more functionality. The trend of distributing machine vision across the entire production line at points before value is added will continue. Neural network techniques seem to have become a key paradigm in machine vision that are used either to correctly segment an image in a wide variety of operational conditions or to classify the detected object. Stereo and 3D-vision applications are also increasingly widespread. Another trend is to utilize machine vision in the non-visible spectrum.

• New product developments will introduce smart camera-based digital imaging systems into existing consumer and industry products, to increase their value and create new products.

• Standards development. One area which may need standardization is the metadata format that facilitates integration and communication between different cameras, sensors and modules in a distributed and augmented video surveillance system. New communication protocols may be needed for better communication between different smart camera products.

Acknowledgements

The authors would like to thank Dr. Xing Zhang from ST Microelectronics and Dr. Julien Epps of National ICT Australia for their many valuable comments and corrections of parts of this paper.

About the Authors – YU SHI is a Senior Researcher with National ICT Australia in Sydney, Australia. He was granted his B.Eng in 1982 by the National University of Defense Technology in Changsha, Hunan, China. He later obtained his M.Eng and PhD in signal processing and biomedical engineering in 1988 and 1992 respectively in Toulouse, France. He also completed post-doctoral research at Oxford Brookes University in England in the late 1990s. His main research interests are in embedded vision systems, FPGA-based design and applications, multimodal user interfaces and web services.

SERGE LICHMAN is a Senior Research Engineer with National ICT Australia in Sydney, Australia. He received M.Eng in Electrical Engineering in 1988 by the Odessa State Polytechnic University in Ukraine. His 12 years of experience in the area of image and signal processing for commercial software and hardware gave him practical skills in full product development life cycles, from research to deployment. His work has led to several publications.