Journal of Electronics and Sensors
Research Article

Smart Cameras in Embedded Systems

Shribala N*, Mary JS*, Mounika N, Manasa P, pragnya K

[1] Department of Electronics Communication Engineering , Bhoj Reddy Engineering college for Women, India

Received: 16 March, 2020; **Accepted**: 20 March, 2020; **Published**: 24 March, 2020

***Corresponding Author:** Shribala N, Department of Electronics Communication Engineering , Bhoj Reddy Engineering college for Women, India . E-mail: shribalanagul71@gmail.com

Introduction

A smart camera performs real-time analysis to recognize scenic elements. Smart cameras are useful in a variety of scenarios: surveillance, medicine, etc. We have built a real-time system for recognizing gestures. Our smart camera uses novel algorithms to recognize gestures based on low-level analysis of body parts as well as hidden Markov models for the moves that comprise the gestures. These algorithms run on a Tri media processor. Our system cans recognize gestures at the rate of 20 frames /second. The camera can also fuse the results of multiple cameras. The smart camera – a whole vision system contained in one neat housing can be used anywhere, in any industry where image processing can be applied. Companies no longer need a cabinet in which to keep all their computing equipment: the computer is housed within the smart camera. In the pharmaceutical industry and in clean rooms – when not even dust is allowed – this can be a big advantage. A single square meter of space can be comparatively very expensive if there is no need for a component rack or cabinet, simply a smart camera, and then this could save a lot of money. In particular, there would not be the usual cabling involved for other vision systems, and set-up is simple. Later in this communication are stated some advantages of using smart cameras or PC-based systems in vision applications. In usual vision systems scenarios, only a small fraction of a picture frame will be the region of interest (ROI). In fact, often no visual image of the ROI is even required. The object of a vision system, after all, is to make a decision: "Is there a blob"? "Where is the blob"? "Is this a defect"?.

Aim of the seminar report

A "smart camera" is basically a video camera coupled to a computer vision system in a tiny package. This communication begins stating the main differences between smart cameras Smart camera architecture is described whereby a combination of an on-board microprocessor and PLD's allow for the embedding of image processing algorithms in the camera.

Motivation for the seminar

Smart cameras are still a pretty new emerging category of devices. The basic idea is connecting your camera to your desktop or fiddling with SD cards is archaic. By running a smartphone OS, you can upload directly to Facebook, Instagram, Drop box, or even use advanced editing apps right on the camera itself. Some even have cellular radios so you can upload over LTE.

Future scope:

To date, exploitation of smart camera technology has been mainly for industrial vision systems, but a cross over is just starting to take place.

Literature of the seminar

To understanding the several concepts in this report several stand books “Smart Cameras in Embedded Systems” by Wayne Wolf, Burak Ozer, TiehanLv, and "A Hierarchical Human Detection System in Compressed and Uncompressed Domains” by Burak Ozer, Wayne Wolf are reffed.

Organization of the seminar report:

In this report the details of the seminar “Smart Cameras in Embedded Systems” have discussed in an elaborate manner.

- Describes the introduction of the seminar Smart Cameras in Embedded Systems and about the aim motivation.
- Describes briefly about Smart Camera.
- Describes briefly about Embedded Systems.
- Describes briefly about Smart Cameras in Embedded Systems.
- Applications of smart cameras.
- Conclusion and some future scopes.

Smart Camera

A smart camera is an integrated machine vision system which, in addition to image capture circuitry, includes a processor, which can extract information from images without need for an external processing unit, and interface devices used to make results available to other devices. A smart camera or "intelligent camera" is a self-contained, standalone vision system with built-in image senses in the housing of an industrial video camera. It contains all necessary communication interfaces, e.g. Ethernet, as well as industry-proof 24V I/O lines for connection to a PLC, actuators, relays or pneumatic valves. It is not necessarily larger than an industrial or surveillance camera. This architecture has the advantage of a more compact volume compared to PC-based vision systems and often achieves lower cost, at the expense of a somewhat simpler (or missing altogether) user interface. Although often used for simpler applications, modern smart cameras can rival PCS in terms of processing power and functionalities. Smart cameras have been marketed since the mid-80s, but only in recent years have they reached widespread use, once technology allowed their size to be reduced while their processing power has reached several thousand MIPS (devices with 1GHz processors and up to 8000MIPS are available as of end of 2006).

Having a dedicated processor in each unit, smart cameras are especially suited for applications where several cameras must operate independently and often a synchronously, or when distributed vision is required (multiple inspection or surveillance points along a production line or with assembly machine) shown in (Figure 1).

Figure (1): Early Smart Cameras

A smart camera usually consists of several (but not necessarily all) of the following components:

- Image sensor (matrix or linear, CCD- or CMOS).
- Image digitization circuitry.
- Image memory.
- Process or (often a DSP or suitably powerful processor).
- Program- and data memory (RAM non-volatile FLASH).
- Communication interface (RS232,Ethernet).
- Olin’s (often opt isolated).
- Lens holder or built in lens (usually C, CS or M-mount).
- Built in illumination device (usually LED).

- Purpose developed real-time operating system (For example VCRT).

A video output (e.g. VGA or SVGA) may be an option for a Smart Camera.

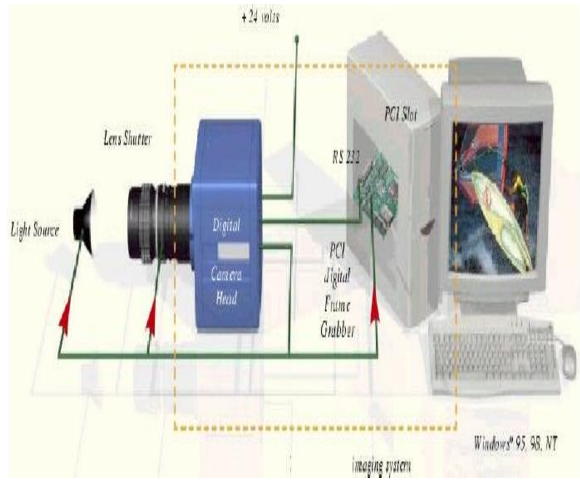
Smart Cameras vs. Standard Smart Vision Systems

The question often comes up as to what is the most appropriate approach to take in implementing a vision system - using a smart camera or using some sort of PC-based approach. There is no question that as the microprocessors, DSPs and FPGAs are getting faster and, therefore, more capable, smart cameras are getting smarter. Therefore, they are a challenge to more "traditional" approaches to vision systems. Significantly, however, "traditional" approaches are also taking advantage of the advances and so, too, are faster and smarter. Traditional approaches usually mean a PC-based implementation. This could be either using a camera with the capability to interface directly to the PC (IEEE 1394/Fire wire, Camera Link, LVDS, USB, etc.), or a system based on a frame grabber or other intelligent image processing board or vision engine that plugs into the PC. In this latter case, more conventional analogy cameras are used as the input device.

A smart camera, on the other hand, is a self-contained unit. It includes the imager as well as the " intelligence" and related I/O capabilities. Because this format resembles the format of many intelligent sensors, these products are often referred to as "vision sensors." However, a vision sensor often has a limited and fixed performance envelope, while a smart camera has more flexibility or tools, inherently capable of being programmed to handle many imaging algorithms and application functions.

A PC-based vision system is generally recognized as having the greatest flexibility and, therefore, capable of handling a wider range of applications.

Figure (2(a)): Standard Smart Vision System



PC-Based Vision Systems Advantages:

The PC-based vision systems advantages include:

Flexibility – The PC offers greater flexibility in the number of options that can be selected. For example one can use a line scan versus an area scan camera with the PC. One can use third party software packages with the PC approach (smart cameras tend to be signal source software).

Power– PC's tend to offer greater power and speed due in large part to the speed of the Intel processors used internally. This power in turn means that PC's are used to handle the "tougher" applications in vision systems.

Smart Cameras Advantages:

The smart cameras advantages include:

- **Cost** – Smart cameras are generally less expensive to purchase and set up than the PC based solution, since they include the camera, lenses, lighting (sometimes), cabling and processing.
- **Simplicity** – Software tools available with smart cameras are of the point-and-click variety and are easiertouse thethose available on PC's. Al gorithms come prepackaged and do not n eed to bedeveloped, thus making the smart camera quicker to setup and use.

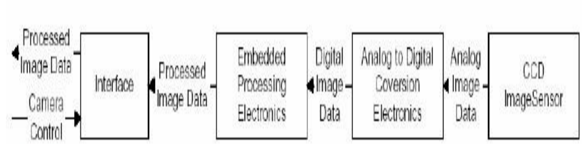
- **Integration**–Given their unified packaging, smart cameras are easier to integrate into the manufacturing environment.
- **Reliability** – With fewer moving components (fans, hard drives) and lower temperatures, smart cameras are more reliable than PC's. In general the performance of the smart camera will continue to increase. This will mean that the smart camera will be used for more difficult applications, slowly displacing the PC approach.

Figure (2(b)): Smart Camera System.

Smart Camera Architecture:

The smart camera presented in this communication reduces the amount of data generated to the 'data of interest' by making use of embedded image processing algorithms. The data of interest might be, for example, defective areas of the product being inspected. Multiple cameras can route their data to a single frame grabber and computer due to the reduction of data stream, thus dramatically reducing system cost and increasing inspection bandwidth capability. This smart camera also makes use of an on-board microprocessor for communication with the inspection systems' host computer and for internal control functions. The following block diagram illustrates the camera architecture.

Figure (2(c)): Smart Camera Architecture Block Diagram



A detailed explanation of the camera architecture follows, starting with the image sensor.

Image Sensor Basics

In this smart camera, a CCD (Charge Coupled Device) image sensor converts photons (light) in to electrons (charge). When photons hit an image sensor, the sensor accumulates electrons. This is called charge integration. The brighter your light source, the more photons available for the sensor to integrate, and the smaller the amount of time required to collect a given amount of light energy Finally, the sensor transfers its aggregate charge to readout registers, which feed each pixel's charge from the image sensor into an output node that converts the charges into voltages. After this transfer and conversion, the voltages are amplified to become the camera's analogy output.

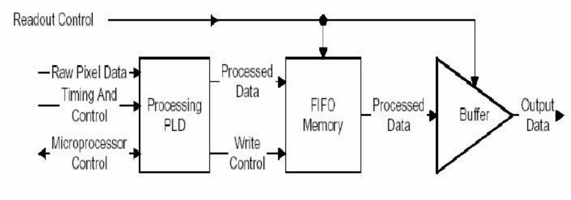
Analog to Digital Conversion Electronics

The analog output of the CCD is converted to a digital output for further processing. The camera presented here sub-divides the CCD analog output into eight channels of 256 pixel element search. Analog to digital conversion is performed at a 20 MHz data rate for each channel thus yielding an effective camera data rate of 160 MHz The digital data is then passed along to the image processing electronics for processing and analysis.

Image Processing Electronics:

Image processing is performed by embedded algorithms on a per channel basis. The following block diagram illustrates the basic processing architecture for each channel.

Figure (2(d)): Image Processing Architecture Block Diagram



The processing algorithm is embedde d in the processing PLD. The microprocessor has a bidirectional path being able to randomly access the algorithm parameters, as well as program anew algorithm into the PLD as required by the user. Raw pixel data and associated timing and control signals are also connected to input pins into the processing PLD. For storage and subsequent readout, algorithm processed data is output along with a write control signal to FIFO memory. 8:1 multiplexing of the data is achieved by using FIFOmemory.Readout control is accomplishe d by the microprocessor/FIFO readout contro l card whose architecture is shown in the (Figure2 (e)).

Figure (2(e)): Microprocessor/FIFO Readout Control Circuit Board Block Diagram

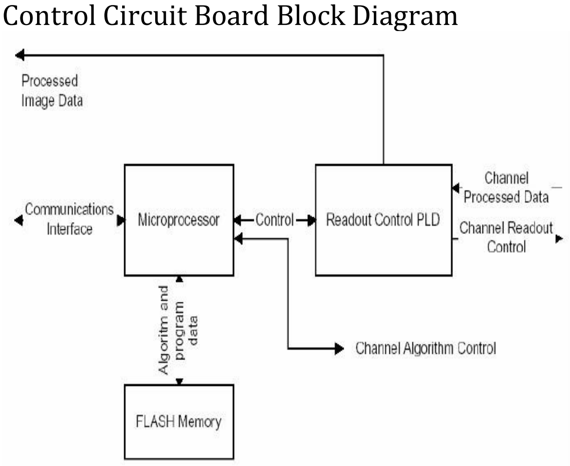


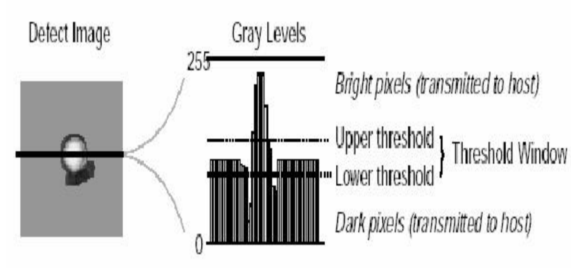
Image Processing Algorithms:

Many types of image processing algorithms can be embedded within the camera, since the video processing modules are completely in-system programmable. As an example, a static grey scale thresholding algorithm is presented below.

Static Grey Scale Thresholding

In static thresholding, an upper and a lower bound are established around what is considered a normal value. Image data that falls within the boundary window is considered normal non-interesting data. Image data that falls either above or below the boundary window is considered data of interest. Considering we are dealing with an 8-bit digital camera, the normal, upper and lower boundary values are seen to be digital numbers (DN) on a scale of 0 to 255 "Grey scale". Imagine that a product is being inspected for defects and the grey scale level of non-defective product is 85 DN, and the upper and lower boundary values have been set to +/- 15 DN. All image data that fell within the bounds of 70 DN to 100 DN would be considered non-Interest in gland would not be transmitted out of the camera. Image data that fell below 70 DN and above 100 DN would be considered interesting and would be transmitted out of the camera. Substantial data reduction is achieved since only some of the data will fall outside of the established boundaries. It is important to note that all of the 'data of interest' is transmitted out of the camera and thus data reduction is achieved where all of the grey scale information is preserved. This type of algorithm is illustrated by the image shown in (Figure 2(f)).

Figure (2(f)): an Example of Static Thresholding



For later display and analysis each pixel must be given an address such that an image can be reconstructed by the frame grabber since an algorithm of this type produces non continuous data. The static thresholding algorithm requires three parameters as follows an upper bound, a lower bound, and a centre value. The determination of the centre value is essential to this type of algorithm, and the acceptable band between the upper and lower bound.

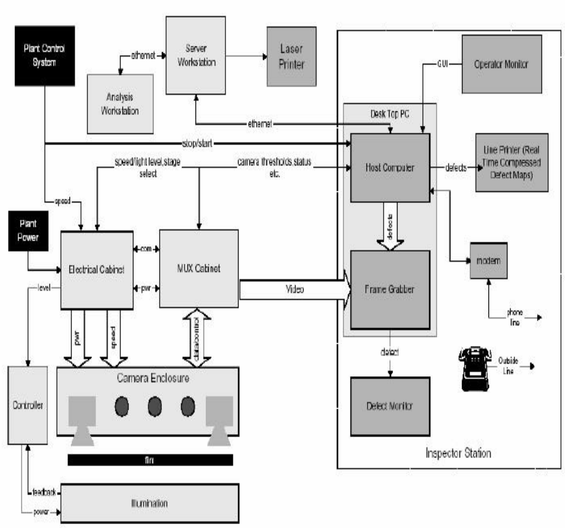
Embedded Image Processing Algorithms

The algorithms are embedded in hardware with a PLD/microcontroller combination and operate at a 20MHz data rate per channel. The effective processing rate is 40MHz because each image processing PLD can process two channels of image data. With dedicated DSP controllers such data processing rates could be difficult to achieve. Microcontroller also can directly control the algorithm without host computer intervention, since it has access to the image data.

Smart Camera System

A vision system for web inspection is presented below where a maximum of twenty 2048 pixel high sensitivity line scan smart cameras are networked together to a single host computer and frame grabber. A block diagram of the system is shown in (Figure 2(g)).

Figure (2(g)): System Block Diagram

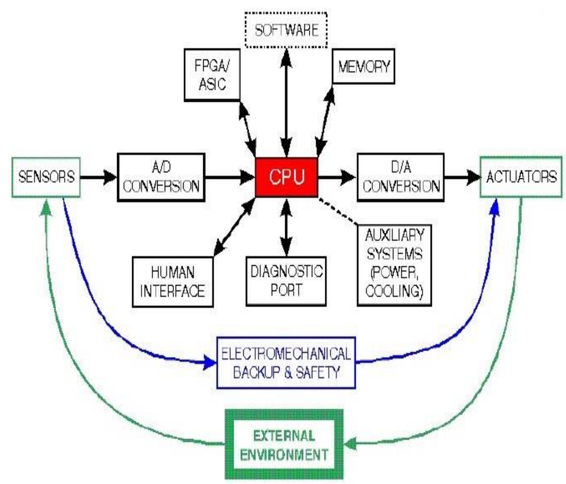


The system shown consists of up to twenty 2048 pixel high sensitivity line scan smart cameras housed within a camera enclosure mounted above the web. Routed through two cabinets are the data, control, and power lines to/from the cameras.

Embedded Systems

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems control many of the common devices in use today. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.Physically, embedded systems range from portable devices such as digital watches andMP4 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants (Figure 3 a).

Figure (3(a)): An Embedded System



Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure. In general embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded computing systems

- Computing systems embedded within electronic devices
- Hard to define. Nearly any computing system other than a desktop computer
- Billions of units produced yearly, versus millions of desktop units
- Perhaps 50 per household and per automobile.

Examples of embedded systems

Embedded systems span all aspects of modern life and there are many examples of their use. Telecommunications systems employ numerous embedded systems from telephone switches for the network to mobile phone sat the end-user. Computer networking uses dedicated router sand network bridges to route data. Consumer electronics include personal digital assistants(PDAs),mp3players, mobile phones ,videogameconsoles,digitalcameras, DVD play ers,GPS receivers, and printers.Manyhousehol d appliances, such asmicrowave ovens,washin g machines and dishwashers,areincluding embedded systems to provide flexibility, efficiency and features. Advanced HVAC syste ms use networked thermostats to more accurately and efficiently control temperature that can change by time of day and season. Home automation uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New air planes contain advanced avionics such as inertial guidance systems and GPS receivers that also have considerable safety requirements. Various electric motors brush less DC motors, induction motor sand DC motors— are using electric/electronic motor controllers. Automobiles, electric vehicles, and hybrid vehicles are increasingly using embedded systems to maximize efficiency and reduce pollution.

Figure (3(b)): PC Engines' embedded board

Other automotive safety systems such as anti-lock braking system(ABS),Electronic Stability Control(ESC/ESP),traction control(T CS) and automatic four-wheel drive. Medical equipmentis continuing to advancewithmore embedded systems for vitalsigns monitoring, electronicstethoscopes'foramplifying sounds, and various medical imaging (PET,SPECT,CT, MRI) for non-invasive internal inspections.In addition to commonly described embedded systems based on small computers, a new class of miniature wireless devices called mot es are quickly gaining popularity as the field of wireless sensor Networking rises. Wireless sensor networking, WSN, makes use of miniaturization made possible by advanced IC design to couple full wireless subsystems to sophisticated sensor ,enabling people and companies to measure a myriad of things in the physical world and act on this information through IT monitoring and control systems. These motes are completely self-contained, and will typically run off a battery source for many years before the batteries need to be changed or charged. An Embedded System is combinations of Hardware and Software that may have some mechanical components to perform specific tasks.

Characteristics of embedded systems

- **Device programmability or manageabl e:** The functioning of a specifically designed hardware part on a smart device can easily be changed by simply changing the software associated with it.
- **Multitasking:**An embedded system empl oyed in a modern refrigerator performs d oor sense andtemperature sense at the same times, which are two functions at the same time.
- **Real time response:** It is the ability of an embedded system to respond to ambient conditions suddenly. That is, a smart TV adjusts picture quality suddenly in response to sudden environmental bright ness variations.
- Embedded systems are designed to do some specific task, rather than be a general- purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.
- Embedded systems are not always standalone devices. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music.
- Embedded system in anautomobile provides a specific function as subsystem of the car itself. The program instructions written for embedded systems are referred to as firmware, and are stored in read-only memory or Flash memory chips. They run with limited computer hardware resources: little memory, small or non-existent keyboard and/or screen.

Smart Cameras in Embedded Systems

The system also provides input and output to the plant control system. Data from the cameras is acquired by the frame grabber, assembled into images, and then transferred to the host computer real time display on the defect monitor, and stored to a database residing on the file server via an Ethernet connection. Subsequent analysis of the data is performed at the analysis work station with analysis software that allows extraction of data from the database for creation of reports. All system and analysis software is multithreaded and provides real time data access and display. Via a modem connection the system is also operable remotely. To ensure smooth and constant illumination of the web the system software also controls the illumination source with a fuzzy logic control scheme. Recent technological advances are enabling a new generation of smart cameras that represent a quantum leap in sophistication. While today's digital cameras capture images, smart cameras capture high-level descriptions of the scene and analyze what they see.These devices could support a wide variet y of applications including human and animal detection, surveillance, motion analysis, and facial identification.

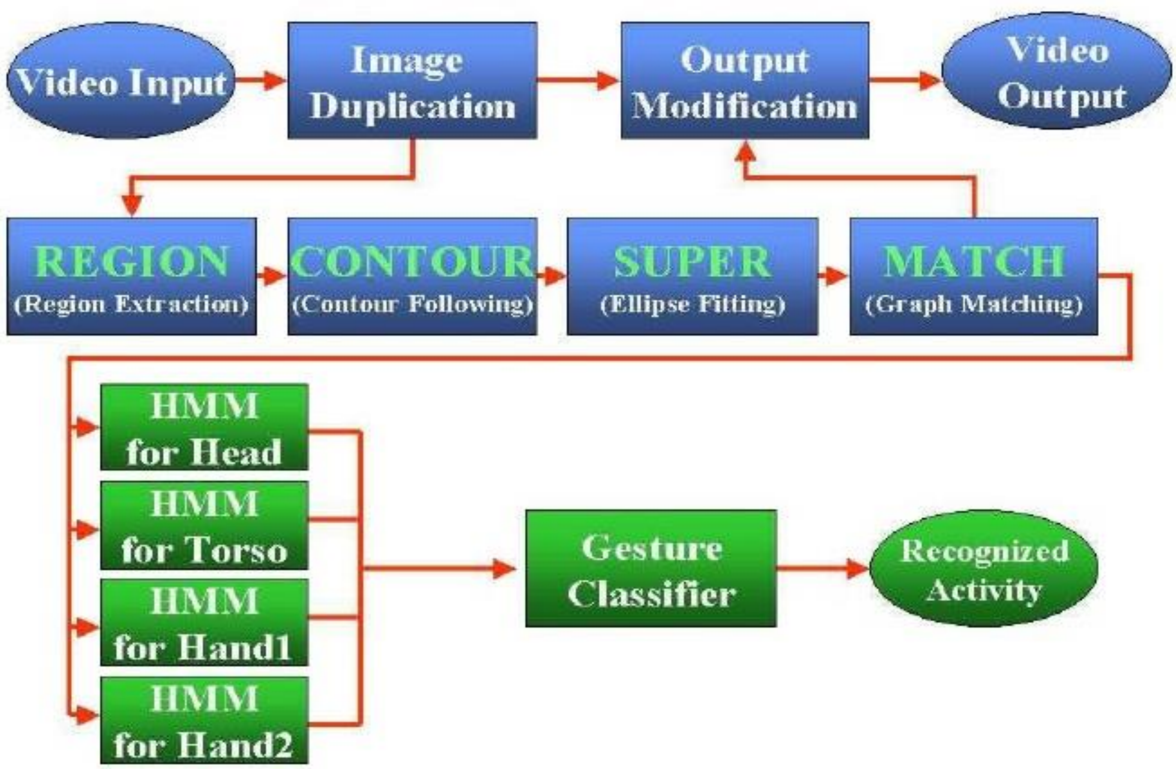
Video processing has an insatiable demand for real-time performance. Fortunately, Moore's law provides an increasing pool of available computing power to apply to real-time analysis. Smart cameras leverage very large-scale integration (VLSI) to provide such analysis in a low-cost, low-power system with substantial memory. Moving well beyond pixel processing and compression, these systems run a wide range of algorithms to extract meaning from streaming video. Because they push the design space in so many dimensions, smart cameras are a leading edge application for embedded system research.

Detection and Recognition Algorithms

Although there are many approaches to real-time video analysis, we chose to focus initially on human gesture recognition—identifying whether a subject is walking, standing, waving his arms, and so on. Because much work remains to be done on this problem, we sought to design an embedded system that can incorporate future algorithms as well as use those we crea ted exclusively for this application. Our algorithms use both low-level and high-level processing. The low-level component identifies different body parts and categorizes their movement in simple terms.

The high-level component, which is application-dependent, uses this information to recognize each body part's action and the person's overall activity based on scenario parameters. Human detection and activity/gesture recognition algorithm has two major parts: Low-level processing (blue blocks in Figure 4) and high-level processing (green blocks in Figure 4)

Figure (4) : Algorithms.



Low-level processing

The system captures images from the video input, which can be either uncompressed or compressed (MPEG and motion JPEG), and applies four different algorithms to detect and identify human body parts.

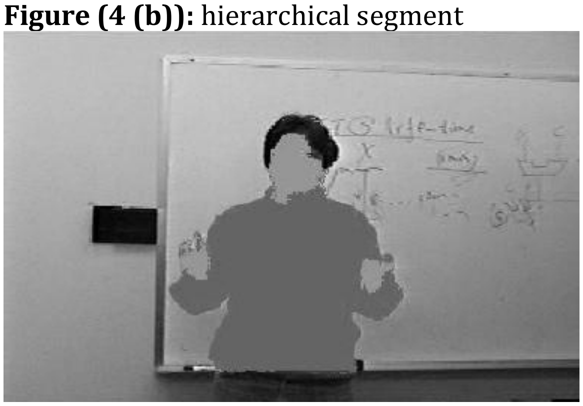
Region extraction

The first algorithm transforms the pixels of an image like that shown in (Figure 4 a), into an M ¥ N bitmap and eliminates the background. It then detects the body part's skin area using a YUV color model with chrominance values down sampled

Figure (4 (a)): transforms the pixels of an image

Next, as (Figure 4 b) illustrates, the algorithm hierarchically segments the frame into skin tone and non-skin-tone regions by extracting foreground regions adja3cent to detected skin areas and combining these segments in a meaningful way.

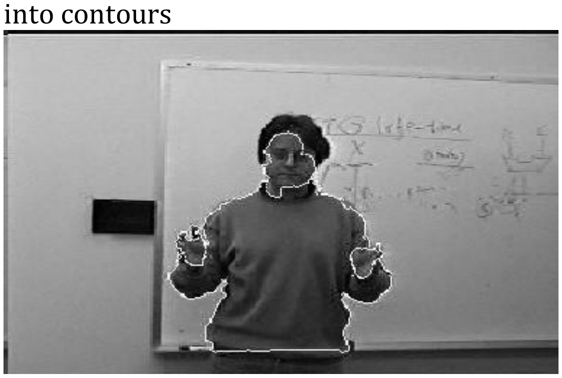
Figure (4 (b)): hierarchical segment



Contour following

The next step in the process, shown in (Figure 4 c), involves linking the separate groups of pixels into contours that geometrically define the regions. This algorithm uses a 3 ¥ 3 filter tofollow the edge of the component in any of eight different directions.

Figure (4(c)): the separate group of pixels into contours



Ellipse fitting

To correct for deformations in image processing caused by clothing, objects in the frame, or some body parts blocking others, an algorithm fits ellipses to the pixel regions as Figure 4.2dshows to provide simplified part attributes. The algorithm uses these parametric surface approximations to compute geometric descriptors for segments such as area, compact ness (circularity), weak perspective invariants, and spatial relationships.

Figure (4(d)): an algorithm fits ellipses to the pixel regions

Graph matching

Each extracted region modeled with ellipses corresponds to a node in a graphical representation of the human body. A piecewise quadratic Bayesian classifier uses the ellipses parameters to compute feature vectors consisting of binary and unary attributes. It then matches these attributes to feature vectors of body parts or meaningful combinations of parts that are computed offline. To expedite the branching process, the algorithm begins with the face, which is generally easiest to detect.

High-level processing

The high-level processing component, which can be adapted to different applications, compares the motion pattern of each body part—described as a spatiotemporal sequence of feature vectors —in a set of frames to the patterns of known postures and gestures and then uses several hidden Markov models in parallel to evaluate the body's overall activity. We use discrete HMMs that can generate eight directional code words that check the up, down, left, right, and circular movement of each body part. Human actions often involve a complex series of movements. We therefore combine each body part's motion pattern with the one immediately following it to generate a new pattern. Using dynamic programming, we calculate the probabilities for the original and combined patterns to identify what the person is doing. Gaps between gestures help indicate the beginning and end of discrete actions. A quadratic Mahalanobis distance classifier combines HMM output with different weights to generate reference models for various gestures. For example, a pointing gesture could be recognized as a command to "go to the next slide" in a smart meeting room or "open the window" in a smart car, whereas a smart secu rity camera might interpret the gesture as suspicious or threatening

To help compensate for occlusion and other image-processing problems, we use two cameras set at a 90-degree angle to each other to capture the best view of the face and other key body parts. We can use high-level information acquired through one view to switch cameras to activate there cognition algorithms using the second camera. Certain actions, such as turning to face another direction or executing a predefined gesture, can also trigger the system to change views Soft-tissue reconstruction. We can use Mat Lab to develop our algorithms. This technical computation and visualization programming environment runs orders of magnitude
more slowly than embedded platform implementations, a speed difference that becomes critical when processing video in real time. We can therefore port our Mat Lab implementation to C code runningonaverylon ginstructionword(VLIW) video processor, wh ich lets us make many architectural measure ments on theapplication and make the necessary optimizations to architect a custom VLSI smart camera.

Requirements

At the development stage, we can evaluate the algorithms according to accuracy and other familiar standards. However, an embedded system has additional real-time requirements:

- **Frame rate:** The system must process a certain amount of frames per second to properly analyze motion and provide useful results. The algorithms we use as well as the platform's computational power determine the achievable frame rate, which can be extremely high in some systems.
- **Latency:** The amount of time it takes to produce a result for a frame is also important because smart cameras will likely be used in closed-loop control systems, where high latency makes it difficult to initiate events in a timely fashion based on action in the video field. Moving to an embedded platform also meant that we have to conserve memory. Looking a head to highly integrated smart cameras; we want to incorporate as little memory in the system as possible to save on both chip area and power consumption. Gratuitous use of memory also often points to inefficient implementation.

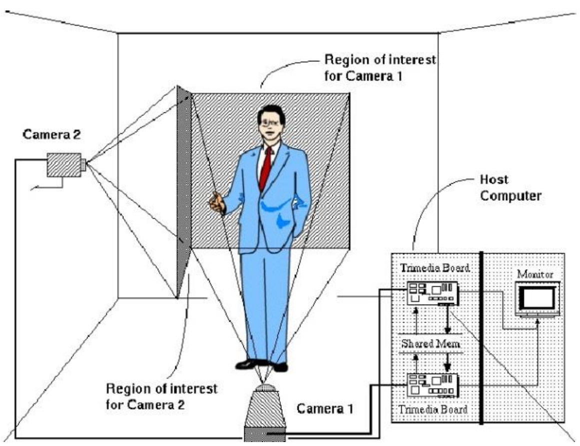
Components

Our development strategy calls for leveraging off-the-shelf components to process video from a standard source in real time, debug algorithms and programs, and connecting multiple smart cameras in a networked system. We use the 100-MHz Philips Tri Media TM-1300 as our video processor. This 32-bit fixed- and floating-point processor features a dedicated image coprocessor, a variable length decoder, an optimizing C/C++ compiler, integrated peripheralsforVLI Wconcurrent realtime input/output, and a ric h set of application library functions includin gMPEG, motion JPEG, and 2D text and graphics.

Test bed Architecture

Our testbed architecture, shown in Figure 4.3, uses two Tri Media boards attached to a host PC for programming support. Each PCI bus board is connected to a Hi8 camera that provides NTSC composite video. Several boards can be plugged into a single computer for simultaneous video operations.

Figure (4(e)) Test bed architecture



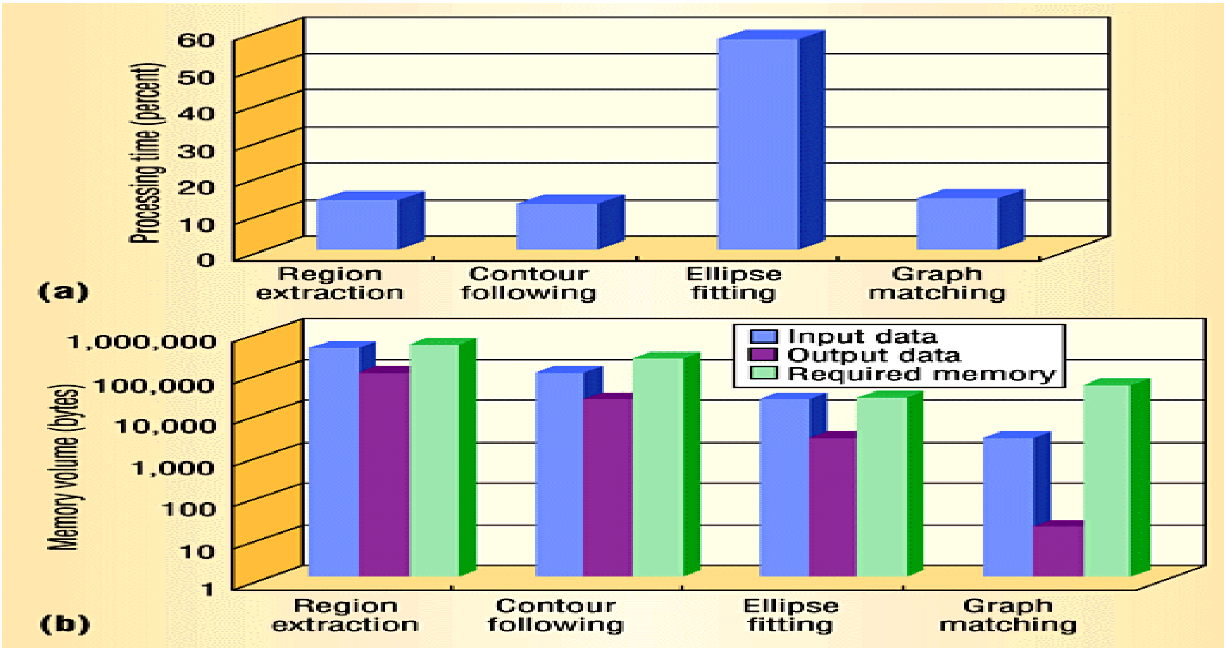
Experiments and Optimizations

As data representation becomes more abstract, input/output data volume decreases. The change in required memory size, however, is less predictable given the complex relationships that can form between abstract data. For example, using six single precision, floating-point parameters to describe 100 ellipses requires only 2.4 Kbytes of memory, but it takes 10 Kbytes to store information about two adjoining ellipses. Based on these early experiments, we optimize our smart camera implementation by apply in techniques to speed up video operations such as substituting new algorithms better suited to real time processing and using Tri Media library routines to replace C level code.

Algorithmic changes

We originally fit super ellipses (generalized ellipses) to contour points, and this was the most time-consuming step. Rather than trying to optimize the code, we decided to use a different algorithm. By replacing the original method developed from principal component analysis withmoment-based initialization, we reduced the Levenber g- Marquardt fitting procedure, thus decreasing the execution time. After converting the original Matlab implementation into C, we performed some experiments to gauge the smart camera system's effectiveness and evaluate bottlenecks. The UN optimized code took, on average, 20.4 million cycles to process one input frame, equal to a rate of 5 frames per second. We first measure the CPU times of each low-level processing step to determine where the cycles were being spent. Microsoft Visual C++ is more suitable for this purpose than the Tri Media compiler because it can collect the running time of each function as well as its sub functions' times.

Figure(4.1(a)): Shows the processing-time distribution of the four body-part-detection algorithms
Figure (4.1(b)): Shows the memory characteristics of each low-level processing stage.



Control-to-data transformation

Increasing the processor's issue width can exploit the high degree of parallelism that region extraction offers. Using a processor with more functional units could thus reduce processing time during this stage. However, contour following, which converts pixels to abstract forms such as lines and ellipses, consumes even more time. The algorithm also operates serially: It finds a region's boundary by looking at a small window of pixels and sequentially moving around the contour; at each clockwise step it must evaluate where to locate the contour's next pixel. While this approach is correct and intuitive, it provides limited ILP. We evaluate all possible directions in parallel and combined the true/false results into a byte, which serves as an index to look up the boundary pixel in a table. We also manipulate the algorithm's control flow structure to furth er increase ILP. These optimizations double t he contour-following stage's running speed.

Disadvantages of smart cameras

A possible disadvantage is the clearly restricted selection of available camera versions with different resolutions. The computing power is limited compared to PC systems due to the compact format. If optics and lighting are integrated into the system and cannot be replaced by standard components, further functional restrictions must be taken into consideration. Typically the system is bound to a special software package provided by the manufacturer. Intelligent embedded PC cameras try to compensate this disadvantage by offering a complete Windows environment for the classic (proprietary) software. Yet, compared to DSP-based intelligent cameras, they are clearly larger.

Applications of Smart Cameras

Smart cameras can in general be used for the same kind of applications where more complexvision systems are used, and can additionally be applied in some applications where volume, pricing or reliability constraints forbid use of bulkier devices and PC's.

Applications

Typical fields of application are:

- Vision-based smart environments.
- Surveillance and tracking applications.
- Applications based on fusion of visual and other sensory data.
- Multi-view vision for Human-Computer Interfaces (HCI).
- 3D scene analysis with distributed sensors.
- Distributed multimedia and gaming applications.
- Automated inspection for quality assurance (detection of defects, flaws, missing parts...)
- Non-contact measurements.
- Part sorting and identification.
- Code reading and verification (barcode, Data Matrix, alphanumeric etc.)
- Web inspection (inspection of continuously flowing materials such as coils, tubes, wires)

For defect detection and dimensional gauging.

- Detection of position and rotation of parts for robot guidance and automated picking
- Unattended surveillance (detection of intruders, fire or smoke detection)
- Biometric recognition and access control (face, finger print, iris recognition)
- Visual sensor networks

Developers can purchase smart cameras and develop their own programs for special, custom made applications, or they can purchase ready-made application software from the camera manufacturer or from third party sources. Custom programs can be developed by programming in various languages (typically C or C++) or by using more intuitive, albeit somewhat less flexible, development tools where existing functionalities (often called tool or blocks) can be connected in a list a sequence or a dimensional flowchart.

Programming is in a much shorter and somewhat easier development process, available also to non-programmers. Smart cameras running software tailored for a single specific application are often called "vision sensors." The term "smart camera" for an image capture system with embedded computing was used by Ret icon for one of their products, by 1977.

Smart Camera Networks

Self-localizing smart camera networks can serve as an enabling technology for a wide range of higher level applications. Here we focus on two applications where the images from the camera systems are used to derive information about the geometric structure of the environment.

Visual Hull Reconstruction

Multi camera systems are commonly used to derive information about the three dimensional structure of a scene. One approach to the reconstruction problem which is particularly well suited to the proposed self-localizing smart camera network is the method of volume intersection which has been employed in various forms by a number of researchers. This method can be used to detect and localize dynamic objects moving through the field of view of the smart camera network. Here a set of stationary cameras are used to observe one or more objects moving through the scene. Simple background subtraction is employed to delineate the portions of the images that correspond to the transient objects. Once this has been accomplished one can interrogate the occupancy of any point in the scene, P, by projecting it into each of the images in turn and determining whether or not it lies within the intersection of the swept regions. This process can be used to produce an approximation for the 3D structure of the transient objects by sampling points in the volume. The results of such an analysis are shown in below Figures.

Figure (5(a)): Background image of a scene

Figure (5(b)): Image with object inserted

Figure (5(c)): Results of the background subtraction operation

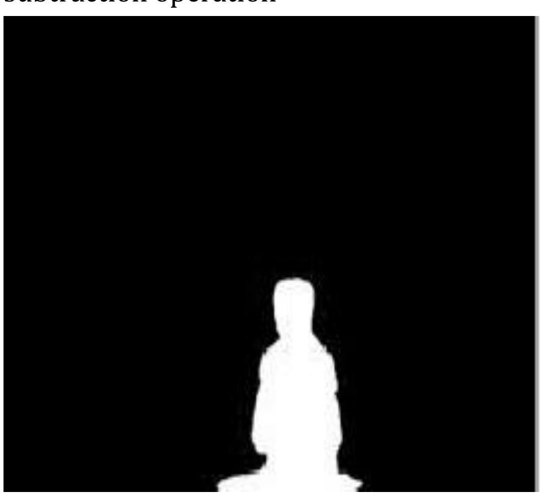
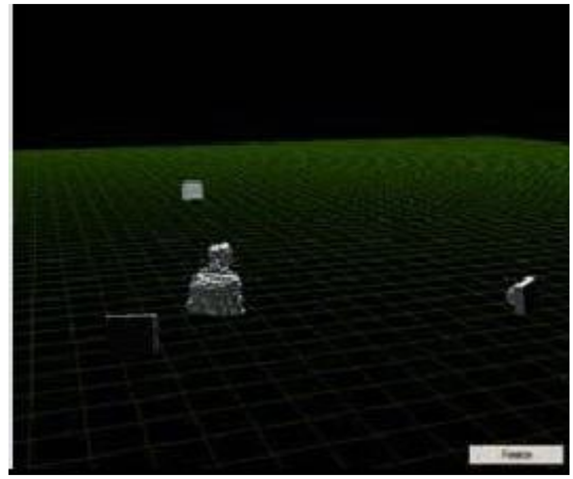


Figure (5(d)): Results of the three smart camera nodes.



Other implementations of this reconstruction scheme involve complex,timeconsuming calib rationOperationsThis implementation, in cont rast, could be be quicklydeployed in an ad-hoc manner and would allow a user to localize and track moving objects such as people, cars or animals as they move through the scene.

Conclusion and Some Future Scopes

Conclusion

The combination of these methods radically improves CPU performance for the application. Optimization boosts the program's frame rate from 5 to 31 frames per second. In addition, latency decreases from about 340 to 40-60 milliseconds per frame. We can add HMMs and other high-level processing parts, and that makes the program now runs at about 25 frames per second. Our board-level system is a critical first step in the design of a highly integrated smart camera. Although the current system is directly useful for some applications, including security and medicine, a VLSI system will enable the development of high volume,embeddedcomp uting products.Because the digital processors and memory use advanced small- fabrication and the sensor requires relatively large pixels to efficiently collect light, it makes sense to design the system as two chips and house them in a multichip module. Separating the sensor and the feature processor also makes sense at the architectural level given the well understood and simple interface between the sensor and the computation engine.

The advantages of leveraging existing sensor technology far outweigh any benefits of using pixel-plane processors until they become more plentiful. However, attaching special-purpose SIMD processors to the multiprocessor can be useful for boundary analysis and other operations. Such accelerators can also save power, which is important given the cost and effort required to deploy multiple cameras, especially in an outdoor setting. High-frame-rate cameras, which are useful for applications ranging from vibration analysis to machinery design, will likely require many specialized processing elements that are fast as well as area efficient. In contrast to common digital cameras, as known for example for taking holiday pictures, SmartCameras are equipped with a computing unit that is used for the analysis of captured images. Recent advances in the research field of Computer Vision allow for the detection of human faces and gestures and for the measurement of distances between objects. In future, Smart Cameras will be able analyse and understand scenes they observe in a cooperative manner. Distributed Smart Cameras can be connected to each other by ad-hoc or infrastructure networks. This Distributed System does not rely on a central control console and has therefore no single point of failure - an important aspect when safety and robustness are concerned. This paper describes a scheme for determining the relative location and orientation of a set of smart camera nodes and sensor modules. The scheme is well suited for implementation on wireless sensor networks since the communication and computational requirements are quite minimal.

Self-localization is a basic capability on which higher level applications can be built. For example, the scheme could be used to survey the location of other sensor motes enabling arrange of location based sensor analyses such as sniper detection, chemical plume detection and target tracking. Further, the ability to automatically localize a set of smart cameras deployed in an ad-hoc manner allows us to apply a number of multi-camera 3D analysis techniques to recover aspects of the 3Dgeometry of a scene from the available imagery. Ultimately we envision being able to construct accurate 3D models of extended environments based on images acquired by a network of inexpensive smart camera systems.

Future Scope

The vision industry is rapidly moving away from the video camera/frame grabber systems of the twentieth century to a new generation of smart-camera-based systems for the 21stcentury. These21st century smart-camera systems will perform real-time, pixel-data extraction and processing operations within the camera at extremely high speeds and at a cost, which is considerably less than required today for comparable capabilities. Eventually, complete vision-processing-systems-on-a-sensor-chipwill is available. Components in smart cameras will undoubtedly change due to the push from semiconductors and new microprocessors coming onto the market. The trends for the next year will be towards megapixel sensors, higher resolution, faster processing power, and colour. A typical standard CCD based camera has a matrix of 480,000 pixels, however the new megapixel cameras offer at least 1,000x1,000 pixels – or 1 million pixels. Some manufacturers already offer cameras of 2 million pixels.

To date, exploitation of smart camera technology has been mainly for industrial vision systems, but a crossover is just starting to take place. Smart camera technology will begin to enter new applications, for example, in the security and access control markets, in the automotive industry, for collision avoidance, and even one day for the toy industry. Even our automobiles may soon be outfitted with miniature eyes. Built into a cruise control system, for instance, such a camera would suddenly alert the driver if it noted a rapidly decelerating vehicle. The cameras could also take the place of the rear view and side-view mirrors, thereby eliminating dangerous blind spots and - in the event of an accident – recording the seconds prior to a collision. Another example would be with intelligent lifts. An office block, with many lifts and floors, may see a lot of people travelling up and down between floors, particularly at high traffic times such as early morning or end of the working day.

Connected with smart camera technology, lifts could be routed on demand, working intelligently, stopping only when there was a pre-set number of passengers waiting at a floor – and missing out a floor if too many people were waiting to meet the maximum capacity of the lift. Looking into the future, we can foresee an infinite number of applications for the smart camera; in fact, as many as there are potential image processing uses. Distributed Smart Cameras can be used for a wide range of applications. To name a few, the following:

- **Assisted Living, Health-Care:** Our society is aging constantly. This leads to a large number of elderly people that need to be cared for. Many elderly move to residential homes because they are frightened of accidents that might occur at home and leave them helpless without the possibility to call for help. Smart Cameras can detect accidents on their own and inform either relatives or a mobile nursing service.
- **Surveillance of large, safety-critical areas (airports, train stations):** Common surveillance systems are prone to errors since human staff in central control rooms can usually not analyse all video data captured by the surveillance cameras. Critical events may pass un noticed and cause great harm that could have been avoided by the use of Smart Cameras. Smart Cameras are able to detect critical events on their own and raise an alarm. Notificati ons about suspiciousincidents can be sent directly from a Smart Camera to a PDA carried by security staff nearby.
- **Industry:** Common sensors used for automation often rely on wired, electromechanical devices. The use of wireless, contactless vision-based devices can help to reduce costs, since maintenance costs of common sensors are high.
- **Retail Analysis:** Internet online shops offer the possibility to analyse the customers' behaviour in detail. Each click and length of stay on pages can be analysed in detail and thereby the customers' satisfaction and the shop's conversion can be increased. For shopping malls and retail stores, this is currently not possible. Smart Cameras can help to analyse customers' behaviour in detail while preserving the customers' privacy by not transmitting video data but operating figures (people counter, duration of stay, routes taken).

Reference

1. Wayne Wolf, Burak Ozer, TiehanLv, “Smart Cameras as High- Performance Embedded Systems”.
2. Burak Ozer, Wayne Wolf, "A Hierarchical Human Detection System in Compressed and Uncompressed Domains”.
3. Tiehan Lv, Burak Ozer, Wayne Wolf, "Sma rt Camera System Design," Invited Paper,I nternational Packet Video Workshop, Pittsburgh, April 2002.
4. M.Nicolescu, G.G. Medioni, "Electronic Pan-Tilt-Zoom: A Solution for Intelligent Room Systems", Proc. IEEE Int'l Conf. Multimedia and Expo, pp. 1581-1584, 2000.
5. Advanced Imaging Europe Magazine: The intelligent camera, October (2002) 12 – 16
6. Wintriss Engineering Corporation: Year 2002 Smart Cameras
7. Smart Cameras vs. PC-based Machine Vision Systems, (2002)
8. Long bottom, D.: Latest Developments in Sensor and Camera Technology, Alrad Instruments Ltd., White paper, pdf (2002).

9. Smart Cameras A complete vision system in a camera body, Vision Components,
10. http://en.wikipedia.org/wiki/Smart_camera
11. http://www.icdsc.org/
12. http://books.google.co.in/books?spell=1&q=Smart+Cameras+as+High+Performance+Embedded+Systems&btnG=Search+Books
13. Camillo J. Taylor Department of Computer and Information Science University of Pennsylvaniacjtaylor@cis.upenn.edu
14. http://www.sra.uni-hannover.de/forschung/projekte/aktuelle-projekte/distributed-smart-cameras-disc.html
15. http://www.scribd.com/doc/3675612/EMBEDDED-SYSTEMS
16. http://www.coreco.com/,
17. http://www.ukiva.org/IPOTMV02Latest.
18. http://www.is.irl.cri.nz/products/smartcam.html
19. Christos Beretas (2018), Security and Privacy in Data Networks. J Electron Sensors; 1(1):1-20.
20. Dutra RF (2018), Silva ACM, Saade J, et al. A carbon ink screen-printed immunoelectrode for Dengue virus NS1protein detection based on photosynthesized amine gold nanoparticles. J Electron Sensors;1 (1):1-16.
21. Jahromi KK (2018), Aktaruzzaman Md, Jalili M Impact of overlapping in the radio coverage areas of multiple Wi-Fi access points on detecting encounters. J Electron Sensors;1 (1):1-18.
22. Aktaş A, Kırçiçek Y (2019), Investigation of Hybrid Renewable Energy Source and Hybrid Energy Storage System. J Electron Sensors; 1 (1): 01-19.
23. Beretas C (2019), Internet of Things (IoT) is Smart Homes and the Risks. J Electron Sensors; 2(1): 1-4.
24. Moursy IA, ElDerini MN, Ahmed MA, (2019) Fault Tolerant Reliable Protocol (FTRP) Performance Evaluation in Wireless Sensor Networks: An Extensitive Study. J Electron Sensors; 2(1): 01-36.
25. Huang X et al., (2020), Electrochemical Determination of Trace Pb (II) by The Modified Glassy Carbon Electrode Multi-Walled Carbon Nanotubes-Nafion-Bi Film. J Electron Sensors; 3(1): 1- 12.
26. Lee H, Yang D, Kim SJ, et al., (2019), Verification of Antenna Sensor with Commercial Handsets By Applying Radiative Calibration Method. J Electron Sensors; 2(1): 01-06.
27. Shribala N, Mary JS, Mounika N, Manasa P et al., (2020), Polymer Electronic System. J Electron Sensors; 3(1): 01-15.
28. Shribala N, Mary JS, Mounika N, Manasa P et al., (2020), Broadband Proximity Coupled Micro Strip Planar Antenna Array For 5g Cellular Applications; 3(1): 01-26.

Citation: Shribala N (2020), Smart Cameras in Embedded Systems.J.Electron Sensor; 3(1): 1-18.
DOI: 10.31829/2689-6958/jes2020-3(1)-111