Embedded System for Biometric Identification

Ahmad Nasir Che Rosli
Universiti Malaysia Perlis
Malaysia

1. Introduction

Biometrics refers to automatic identification of a person based on his or her physiological or behavioral characteristics which provide a reliable and secure user authentication for the increased security requirements of our personal information compared to traditional identification methods such as passwords and PINs (Jain et al., 2000). Organizations are looking to automate identity authentication systems to improve customer satisfaction and operating efficiency as well as to save critical resources due to the fact that identity fraud in welfare disbursements, credit card transactions, cellular phone calls, and ATM withdrawals totals over $6 billion each year (Jain et al., 1998). Furthermore, as people become more connected electronically, the ability to achieve a highly accurate automatic personal identification system is substantially more critical. Enormous change has occurred in the world of embedded systems driven by the advancement on the integrated circuit technology and the availability of open source. This has opened new challenges and development of advanced embedded system. This scenario is manifested in the appearance of sophisticated new products such as PDAs and cell phones and by the continual increase in the amount of resources that can be packed into a small form factor which requires significant high end skills and knowledge. More people are gearing up to acquire advanced skills and knowledge to keep abreast of the technologies to build advanced embedded system using available Single Board Computer (SBC) with 32 bit architectures.

The newer generation of embedded systems can capitalize on embedding a full-featured operating system such as GNU/Linux OS. This facilitate embedded system with a wide selection of capabilities from which to choose inclusive of all the standard IO and built in wireless Internet connectivity by providing TCP/IP stack. Only a few years ago, embedded operating systems were typically found only at the high end of the embedded system spectrum (Richard, 2004). One of the strengths of GNU/Linux OS is that it supports many processor architectures, thus enabling engineers to choose from varieties of processors available in the market. GNU/Linux OS is therefore seen as the obvious candidate for various embedded applications. More embedded system companies development comes with SDK which consists of open source GNU C compiler. This chapter demonstrates the idea of using an embedded system for biometric identification from hardware and software perspective.

2. Biometric Identification

Biometrics is the measurement of biological data (Jain et al., 1998). The term biometrics is commonly used today to refer to the science of identifying people using physiological features (Ratha et al., 2001). Since many physiological and behavioral characteristics are distinctive to each individual, biometrics provides a more reliable and capable system of authentication than the traditional authentication systems. Human physiological or behavioral characteristics that can be used as biometric characteristics are universality, distinctiveness, permanence and collectability (Jain et al., 2000, 2004; Garcia et al., 2003). A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database (Jain et al., 2004). A practical biometric system should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, and be sufficiently robust to various fraudulent methods and attack on the system. A biometric system can operate either in *verification* mode or *identification* mode depending on the application context.

In the verification mode, the system validates a person's identity by comparing the captured biometric data with her own biometric template(s) stored system database such as via a PIN (Personal Identification Number), a user name, a smart card, etc., and the system conducts a one-to one comparison to determine whether the claim is true or not. In the identification mode, the system recognizes an individual by searching the templates of all the users in the database for a match. Therefore, the system conducts a one-to-many comparison to establish an individual's identity without the subject having to claim an identity. The verification problem may be formally posed as follows: given an input feature vector X_Q (extracted from the biometric data) and a claimed identity I, determine if (I, X_Q) belongs to class w_1 or w_2, where w_1 indicates that the claim is true (a genuine user) and w_2 indicates that the claim is false (an impostor). Typically, X_Q is matched against X_I, the biometric template corresponding to user I, to determine its category. Thus,

$$(I, X_Q) \in \begin{cases} w_1 & \text{if } S(X_Q, X_I) \geq t, \\ w_2 & \text{otherwise} \end{cases} \tag{1}$$

where S is the function that measures the similarity between feature vectors X_Q and X_I, and t is a predefined threshold. The value $S (X_Q, X_I)$ is termed as a similarity or matching score between the biometric measurements of the user and the claimed identity. Therefore, every claimed identity is classified into w_1 or w_2 based on the variables X_Q, I, X_I and t, and the function S. Note that biometric measurements (e.g., fingerprints) of the same individual taken at different times are almost never identical. This is the reason for introducing the threshold t. The identification problem, on the other hand, may be stated as follows: given an input feature vector X_Q, determine the identity I_k, $k \in \{1, 2 \dots N, N+1\}$. Here I_1, I_2, …, I_N are the identities enrolled in the system and I_{N+1} indicates the reject case where no suitable identity can be determined for the user. Hence,

$$X_Q \in \begin{cases} I_k & \text{if } \max_k \{S(X_Q, X_{I_k})\} \geq t, k = 1, 2, \dots, N, \\ I_{N+1} & \text{otherwise} \end{cases} \tag{2}$$

where X_{T_k} is the biometric template corresponding to identity I_k, and t is a predefined threshold.

A biometric system is designed using the following four main modules: sensor module, feature extraction module, matcher module and system database module. The sensor module captures the biometric data of an individual such as a camera to capture a person face image for face biometric. The feature extraction module is a very important process where the acquired biometric data is processed to extract a set of salient or discriminatory features. An example is where the position and orientation of face image are extracted in the feature extraction module of a face-based biometric system. The matcher module ensures that the features during recognition are compared against the stored templates to generate matching scores. For example, in the matching module of a face-based biometric system, the number of matching minutiae between the input and the template face images is determined and a matching score is reported. The matcher module also encapsulates a decision making module in which a user's claimed identity is confirmed (verification) or a user's identity is established (identification) based on the matching score.

The system database module is used by the biometric system to store the biometric templates of the enrolled users. The enrollment module is responsible for enrolling individuals into the biometric system database. During the enrollment phase, the biometric characteristic of an individual is first scanned by a biometric reader to produce a digital representation (feature values) of the characteristic. The data captured during the enrollment process may or may not be supervised by a human depending on the application. A quality check is generally performed to ensure that the acquired sample can be reliably processed by successive stages. In order to facilitate matching, the input digital representation is further processed by a feature extractor to generate a compact but expressive representation called a template. Depending on the application, the template may be stored in the central database of the biometric system or be recorded on a smart card issued to the individual. Usually, multiple templates of an individual are stored to account for variations observed in the biometric trait and the templates in the database may be updated over time.

3. Comparison of Biometric Technologies

A number of biometric characteristics exist and are in use in various applications. Each biometric has its strengths and weaknesses, and the choice depends on the application. No single biometric is expected to effectively meet the requirements of all the applications. In other words, no biometric is "optimal". The match between a specific biometric and an application is determined depending upon the operational mode of the application and the properties of the biometric characteristic. Any human physiological or behavioral characteristic can be used as a biometric characteristic as long as it satisfies the requirements such as *universality* where each person posseses a characteristic; *distinctiveness* i.e. any two persons should be sufficiently different in term of the characteristic; *permanence*, where the characteristic should neither change nor be alterable; *collectability,* the characteristic is easily quantifiable; *performance*, which refers to the achievable recognition accuracy and speed, the robustness, as well as its resource requirements and operational or environmental factors

that affect its accuracy and speed; *acceptability* or the extent people are willing to accept for a particular biometric identifier in their daily lives; and *circumvention*, which reflects how easily the system can be fooled using fraudulent methods.

Biometric Identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial Thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Gait	M	L	L	H	L	H	M
Hand Geometry	M	M	M	H	M	M	M
Hand Vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Odor	H	H	H	L	L	M	L
Palmprint	M	H	H	M	H	M	M
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

Table 1. Comparison of various biometric technologies based on the perception of the authors (Jain et al., 2000, 2004; Garcia et al., 2003). H-high, M-medium, L-low.

A brief comparison of various biometric techniques based on the seven factors is provided in Table 1. The applicability of a specific biometric technique depends heavily on the requirements of the application domain. No single technique can outperform all the others in all operational environments. In this sense, each biometric technique is admissible and there is no optimal biometric characteristic. For example, it is well known that both the fingerprint-based techniques are more accurate than the voice-based technique. However, in a tele-banking application, the voice-based technique may be preferred since it can be integrated seamlessly into the existing telephone system. Biometric-based systems also have some limitations that may have adverse implications for the security of a system. While some of the limitations of biometrics can be overcome with the evolution of biometric technology and a careful system design, it is important to understand that foolproof personal recognition systems simply do not exist and perhaps, never will. Security is a risk management strategy that identifies, controls, eliminates, or minimizes uncertain events that may adversely affect system resources and information assets. The security level of a system

depends on the requirements (threat model) of an application and the cost-benefit analysis. The properly implemented biometric systems are effective deterrents to perpetrators.

There are a number of privacy concerns raised on the use of biometrics. A sound tradeoff between security and privacy may be necessary; collective accountability/acceptability standards can only be enforced through common legislation. Biometrics provides tools to enforce accountable logs of system transactions and to protect an individual's right to privacy. As biometric technology matures, there will be an increasing interaction among the market, technology, and the applications. This interaction will be influenced by the added value of the technology, user acceptance, and the credibility of the service provider. It is too early to predict where and how biometric technology would evolve and get embedded in which applications. But it is certain that biometric-based recognition will have a profound influence on the way we conduct our daily business.

4. Face Recognition

Face recognition is an important research problem spanning numerous fields and disciplines and one of the most successful applications of image analysis and understanding. This is due to numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, and surveillance system. Face recognition is a fundamental human behaviors that is essential for effective communications and interactions among people (Tolba et al., 2005). A formal method of classifying faces was first proposed by Galton (1888). The author proposed collecting facial profiles as curves, finding their norm, and then classifying other profiles by their deviations from the norm. This classification is multi-modal, i.e. resulting in a vector of independent measures that could be compared with other vectors in a database. Progress has advanced to the point that face recognition systems are being demonstrated in real-world settings (Zaho, 1999). The rapid development of face recognition is due to a combination of factors--active development of algorithms, the availability of large databases of facial images, and a method for evaluating the performance of face recognition algorithms.

Face recognition is a biometric identification technology which uses automated methods to verify or recognize the identity of a person based on his/her physiological characteristics. A general statement of the problem of face recognition system can be classified as a process to identify or verify one or more persons in the static images or video images of a scene by comparing with faces stored in database (Zhao et al., 2003). Available collateral information such as race, age, gender, facial expression, or speech may be used in narrowing the search (enhancing recognition). Face recognition starts with the detection of face patterns in sometimes cluttered scenes, proceeds by normalizing the face images to account for geometrical and illumination changes, possibly using information about the location and appearance of facial landmarks, identifies the faces using appropriate classification algorithms, and post processes the results using model-based schemes and logistic feedback (Chellappa et al., 1995). In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

All face recognition algorithms consist of two major parts (Tolba et al., 2005): (1) face detection and normalization; and (2) face identification. Algorithms that consist of both parts are referred to as fully automatic algorithms and those that consist of only the second part are called partially automatic algorithms. Partially automatic algorithms are given a facial image and the coordinates of the center of the eyes. Fully automatic algorithms are only given facial images. Face recognition has recently received significant attention, especially during the past few years (Zhao et al., 2003), which is shown by the emergence of face recognition conferences such as the International Conference on Audio and Video-Based Authentication (AVBPA) since 1997 and the International Conference on Automatic Face and Gesture Recognition (AFGR) since 1995, systematic empirical evaluations of face recognition technique (FRT), including the FERET (Phillips et al., 1998), (Phillips et al., 2000),(Rizvi et al., 1998), FRVT 2000 (Blackburn et al., 2001), FRVT 2002 (Phillips et al., 2003) and XM2VTS (Messer et al., 1999) protocols, and many commercially available systems.

5. Hardware Platforms for Embedded Systems

An embedded system has been around for over a decade and enormous change has occurred since then. In the early embedded system application, limitations in component choice resulted in functional limitations. Most embedded systems were run with relatively simple 8-bit microcontrollers. Until recently, the vast majority of these embedded systems used 8- and 16-bit microprocessors, requiring little in the way of sophisticated software development tools, including an Operating System (OS). But the breaking of the $5 threshold for 32-bit processors is now driving an explosion in high-volume embedded applications (Stepner et al., 1999). A new trend towards integrating a full system on- a-chip (SOC) promises a further dramatic expansion for 32- and 64-bit embedded applications. The traditional small, narrowly focused embedded systems retain their significant presence, but these newer arrivals can capitalize on embedding a full-featured operating system, especially Linux OS (Badlishah et al., 2006a). These embedded systems are ubiquitously used to capture, store, manipulate, and access data of a sensitive nature (e.g. personal appliances such as cell phones, PDAs, smart card, portable storage devices), or perform safety-critical functions (e.g. automotive and aviation electronics, medical appliances) (Aaraj et al., 2006).

The integration of embedded computing is a hard task and it is difficult to integrate in both software and hardware. A strong effort by the scientific and industrial community has taken place to overcome this complex issue by splitting up the complex system in different smaller parts with very specific purposes. Ramamritham and Arya (2003) define that embedded applications are characterized by a number of issues: control, sequencing, signal processing and resource management. Tan et al. (2003) describe how energy consumption has become a major focus in embedded systems research and there has been a move from hardware-oriented low energy design techniques to energy-efficient embedded software design. It is known that the Operating System (OS) has a significant impact on the system energy consumption.

Al-Ali et al. (2003) propose a small system for measuring blood pressure and other variables in a patient monitoring device. Kroll et al. (2003) show the use of more complex solution for

medical imaging by using Java on embedded systems in order to incorporate cryptographic algorithms specified in the DICOM standard1. Lamberti and Demartini (2003) propose a design and development of low-cost homecare architecture for remote patient telemetry based on Java software and an embedded computer. The researchers prove the concept by using new approaches like Personal Digital Assistants and WAP enabled GSM/GPRS mobile phones for real-time monitoring of ECGs. In experimental nuclear science there is a high presence of embedded systems research for instrumentation (Deb et al., 2000; Dufey et al., 2000; Fryer, 1998; Gori et al., 1999). In other application, as reviewed in Baber and Baumann (2002), human interaction with embedded technology (in the wearable sense) is considered. In this paper, the authors opine that the Human-Computer Interaction (HCI) will move away from the desktop to be merged into the rest of daily activities.

Product	Manufacturer	Operating System	Processor	Key Features
Psion Series 5 (Handheld)	Psion PLC, London	EPOC32	ARM 7100	Touch-typable keyboard, long battery life
Palm III (Palm device)	3Com Corp., Santa Clara, Claif	Palm OS	Motorola 68328	Longest battery life, very lightweight
Communicator 9110 (Smart Phone)	Nokia Mobile Phones, Espoo, Finland	GEOS 3.0	AMD 486	Portable Digital GSM
IS630 screen phone (Smart Phone)	Philips Consumer Communications LP, Murray Hill, N.J.	Inferno / Personal-Java	Digital Strong ARM 1100	Desktop Unit, IR Keyboard, color display, two PCCard slots, 28.8kbps modem
ICES (In car System)	Visteon Automotive Systems (a Ford Motor enterprise), Dearborn, Mich.	Windows CE 2.0	Intel Pentium	Voice recognition and text-to-speech capability, traffic conditions, navigation, cell phone, rear-seat movies...

Table 2. Information appliances (Comerford, 1998)

Some requirements of an embedded system are different to those which are required for a desktop computer--the system has to be responsive, a considerable design effort is given to system testability (doing a proper debug can be difficult without display or keyboard), there are strong reliability and stability requirements and memory space is limited. On the other hand, special tools are required to program these devices, power considerations are sometimes critical, but high throughput may be needed. And they have always a cost oriented design. An embedded system is considered a computer system hidden inside another product with other goals that being a general purpose computer system. Microprocessor-based cores like Intel x86 family are slowly migrating to embedded systems and are becoming cost-competitive against the alternatives already existent in the market. This fact is provoking dramatic changes in our society as cleverer and faster embedded computer systems are reaching consumer market. These devices are changing the way in which we communicate with each other (mobile telephony) via the addition of high efficiency audio-video encoding/decoding algorithms. These algorithms can be

implemented in cheap (and complex) telephone terminals that optimize bandwidth in such a way that is cost effective enough to sell personal video communication at consumer market.

Perera et al. (2003) show that hardware platforms used at research level and even present market are really varied but there is a clear evolution towards high end profile µProcessors for portable and embedded computing. A study on information appliances is found in Table 2 where it is remarkable that microprocessors from x86 architecture like AMD 486 that were formerly used as the CPU of a desktop computer some years ago are now used as embedded processors in devices like Communicator 9110 from Nokia. This has caused manufacturers to provide the market with different kinds of embedded appliances that are actually full computers (in the sense of CPU, Data Storage Support, Memory, Serial & Parallel ports, Network devices, Data acquisition, etc.). Recently even some companies have begun manufacturing systems based on the so called system-on-chip (SoC from now on), where all CPU peripherals are not included in a chip-set mounted on the same printed circuit board but integrated in the same dice. The migration from 8- to 16- to 32-bit devices is helping the addition of more advanced technologies into consumer markets. From a pattern recognition view this can be noticed in handwriting recognition in PDAs, voice/speech recognition, biometric systems for security, and others. These techniques can be applied into industrial market as well.

6. Image Acquisition and Processing in Embedded Device

A variety of smart camera architecture designed in academia and industry exists today as stated in Bramberger et al (2006) and Wolf et al. (2002). Fleck et al. (2007) suggested that all smart cameras system is the combination of a sensor, an embedded processing unit and a connection, which is nowadays often a network unit. The embedded processing unit can be classified in DSPs, general purpose processors, FPGAs, and a combination thereof. More people are doing research on Linux running embedded on the smart camera. There exist several projects which also focus on the integration of image acquisition and image processing in a single embedded device. Fleck and Straßer (2005) present a particle filter algorithm for tracking objects in the field of view of a single camera. They used a commercially available camera which comprised a CCD image sensor, a Xilinx FPGA for low-level image processing and a Motorola PowerPC CPU. They also implemented a multi-camera tracking (Fleck and Straßer ,2006) using the particle filter tracking algorithm. However, in this work, the handover between cameras is managed by a central server node. Cao et al. (2005) proposed an image sensor mote architecture, in which an FPGA connects to a VGA (640x 480 pixels) CMOS imager to carry out image acquisition and compression. An ARM7 microcontroller processes image further and communicates to neighbouring motes via an ultra-low-power-transceiver.

Rahimi et al. (2005) suggested another powerful image sensor mote, which combines Agilent Technologies' Cyclops with Crossbow's Mica2 mote. Cyclops was developed as an add-on CIF (320x240 pixel) CMOS camera module board, which hosts an on-board 8-bit microcontroller and 64 Kbytes of static and 512 Kbytes of flash memory for pixel-level processing and storage. Oliveira et al. (2006) presented a smart camera mote architecture

that uses an FPGA as its central processing unit, a VGA CMOS imager, and 10 Mbytes of static and 64 Mbytes of flash memory to perform early vision. Downes et al. (2006) introduced mote architecture with minimal component count, which deploys an ARM7 microcontroller as its core, 2 Mbytes flash memory, and a 2.4 GHz IEEE 802.15.4 radio. Velipasalar et al. (2006) described a PC based decentralized multi-camera system for multi-object tracking using a peer-to-peer infrastructure. Each camera identifies moving objects and follows their track. When a new object is identified, the camera issues a labelling request containing a description of the object. If the object is known by another camera, it replies the label of the object; otherwise a new label is assigned which results in a consistent labelling over multiple cameras. Rowe et al. (2005) promoted a low cost embedded vision system. The aim of this project was the development of a small camera with integrated image processing. Due to the limited memory and computing resources, only low-level image processing like threshold and filtering were possible. The image processing algorithm could not be modified during runtime because it was integrated into the processor's firmware.

Agent systems have also been used as a form of abstraction in multi-camera applications. Remagnino et al. (2001) described the usage of agents in visual surveillance systems. An agent based framework is used to accomplish scene understanding. Abreu et al. (2000) presented Monitorix, a video-based multi-agent traffic surveillance system based on PCs. Agents are used as representatives in different layers of abstraction. Quaritsch et al. (2006) presented a decentralized solution for tracking objects across multiple embedded smart cameras that combine video sensing, processing and communication on a single embedded device which is equipped with a multi-processor computation and communication infrastructure. Each object of interest has a corresponding tracking instance which is represented by a mobile agent. Hengstler and Aghajan (2006) introduced energy-efficient smart camera mote architecture with intelligent surveillance. This is a low-resolution stereo vision system continuously determines position, range, and size of moving object entering its field of view. Kleihorst et al. (2006) introduce a wireless smart camera based on a SIMD video-analysis processor and an 8051 microcontroller as a local host. Williams et al. (2006) described the design and implementation of two distributed smart camera applications i.e. a fall detector and an object finder. Fleck et al. (2007) propose network-enabled smart cameras for probabilistic tracking. The smart cameras' tracking results are embedded in an integrated 3D environment as live textures and can be viewed from arbitrary perspectives.

7. Single Board Computer (SBC)

Single Board Computers (SBCs) have changed dramatically over the years. Early microcomputer typically consisted of circuit board which implemented the central processing unit (CPU), memory, disk controllers and serial/parallel port functions. These microcomputers are used for data acquisition, process control, and R&D projects, but are generally too bulky to be used as the intelligence embedded within devices (LinuxDevices, n.d.). Advancement in the density, complexity and capability of the silicon improved the choice and selection methodology for SBCs. Today, software, board size, and time-to-market are the key decision factors in addition to just the power and speed of the CPU. Historically, the initial SBC structure was a simple extension of the common bus architecture used by the

microprocessor. It had an onboard local bus and off-board expansion bus. Early SBCs could only support a minimum number of functions on a single board. Therefore, initial SBC specification standards focused on memory and I/O expansion by means of multiple boards connected via a backplane bus or mezzanine. However, as the SBC market has evolved and matured, the backplane bus importance has diminished.

The first industrial microprocessor based SBC standard was Intel's Multibus I introduced in the late 70's (Robert, n.d.). It was optimized for Intel's 80xx processor family. In the early 1980's integrated circuit (IC) technology had advanced to where functions that occupied entire circuit boards could be crammed into single "large-scale integrated" (LSI) logic chips (Badlishah, 2006a). The result of the semiconductor advances was that it was possible to increase the functional density on the boards while decreasing cost and increasing reliability. Instead of a system requiring multiple boards, a complete microcomputer system is implemented on a single board. Three technical developments will impact the use of single board computers in industrial automation in the near term. They are flat panel display technology, network-based computing, and Linux. There are a wide range of architectures that are being used to develop an embedded system. In general, embedded systems can be divided into three classes i.e. Small Scale Embedded Systems, Medium Scale Embedded Systems and Sophisticated Embedded Systems (Badlishah et al., 2006b). Small scale embedded systems have a less hardware and software complexities. They are designed with a single 8- or 16-bit micro-controller and involve board level design. Examples: 8051, 68HC05. Medium scale embedded systems have both hardware and software complexities. They are designed with a single or few 16- or 32-bit micro-controller or Reduced Instructions Set Computer (RISCs). Sophisticated embedded systems have an enormous hardware and software complexities. Besides they may need scalable processor or configurable processor. The TCP/IP stacking and network driver are also implemented in the hardware. Examples: PowerPC, ARM7 and Intel 80960CA.

Generally there are four categories of embedded systems which are stand-alone embedded systems, real-time embedded systems, networked appliances and mobile devices. Table 3 lists a few of the embedded system built using different architectures (Richard, n.d.). There are a number of reasons developers should choose to use SBC for development such as speed development, low development cost, increasing clock speed and availability of GNU/Linux(Badlishah et al., 2006a). An embedded system designed from scratch requires that boards be designed, fabricated, and debugged. The software must be tested and debugged on the target system. In addition, high speed buses like PCI take more design effort to get it right; instead, SBC boards had someone else did the job of making it works. Embedded systems based on SBC require no costly board design/fabrication/debug cycles. Standard PC Tools are usually used for software development, eliminating the need to purchase emulators. As product development cycles get shorter, there is an incentive to buy proven, off-the-shelf components. Another factor is the increasing clock speed of the hardware which passes to GHz range. Due to the GNU/Linux free kernel that is available for most of CPU architecture, it makes the application development much easier. Even the source code of some of the applications is in the Internet.

Product	Platform (Microprocessor)
Vendo V-MAX 720 vending machine	8-bit Motorola 68HC11
Sonicare Plus toothbrush:	8-bit Zilog Z8
Miele dishwashers	8-bit Motorola 68HC05
NASA's Mars Sojourner Rover	8-bit Intel 80C85
Garmin StreetPilot GPS Receiver	16-bit
Palm Vx handheld	32-bit Motorola Dragonball EZ
Motorola i1000plus iDEN Multi-Service Digital Phone	Motorola 32-bit MCORE
Rio 800 MP3 Player	32-bit RISC
RCA RC5400P DVD player	32-bit RISC
Sony Aibo ERS-110 Robotic Dog	64-bit MIPS RIS

Table 3. Embedded System with different hardware platform (Richard, n.d.)

Some embedded designs can still be accomplished using processor with clock rates in the low MHz range. However, as clock rates go up and the development costs follow, more companies concentrate their effort on the hardware and software that makes their product unique. Off-the-Shelf CPUs, Ethernet boards, and similar components parts are treated as commodity parts, which they are. So why assign an engineer to spend three months developing a board that looks and works like a hundred other identical designs? Single Board Computer (SBC) is one type of embedded system technology widely used for recent years. SBC can perform specific tasks like computer as it has a processor, RAM, hard disk and OS or languages. Many applications have been developed for current and future technology as described in (Dan, 2003; Wiencke, 2006; Wood, 2006).

8. System Overview

This research focus on the development and implementation of embedded system for biometric identification based on iris detection using SBC and GNU/Linux which enables the replacement of traditional techniques of authentication system for security such as smart card reader system. This system uses *Face Reader* to acquire face image and performs the image preprocessing process to extract a facial features of a person's face for biometric identification purposes. The approach proposed was the use of an embedded system (SBC) for controlling the external devices such as Universal Serial Bus (USB) web camera, LCD panel and matrix keypad and connectivity. The control was executed via ANSI-C software coded on top of an open source operating system (GNU/Linux).

The software code is portable to a desktop system for integration with other software components such as biometric identification software. Only changes in acquisition devices such camera and keypad module is required in order to perform the task. The software code is portable to a small embedded system without the need of the specific SBC or without the use of SBC based system. The software works in any platform where Linux kernel has been ported. The software code is written regardless any limitation of the hardware platform such as slow processing speed and low image quality captured by the camera.

9. Hardware Design

Face Reader hardware design is composed of a SBC as the main component. Other components such as the Universal Serial Bus (USB) webcam, LCD panel, Compact Flash Card, PCMCIA Wireless Network Card and matrix keypad are attached to the SBC. Figure 1(a) shows the necessary components used for the proposed system. The integration and configuration of the hardware components constituted a physical model of a *Face Reader*. Figure 1(b) shows the illustration of a *Face Reader* model. *Face Reader* is responsible for testing initialization and image capturing processes. It is a compact SBC unit with a USB webcam mounted on top of it. The LCD Panel provides the mechanism of input prompting, system status information display and processing results. It is used to communicate with users. The keypad, which is mounted in front of the unit, is used by the user to key in their user ID or presses the instructed key by referring to the LCD. Getting the system to function appropriately involves the installation of appropriate device driver module to the operating system (OS) by considering the version of Linux kernel and libraries on board.

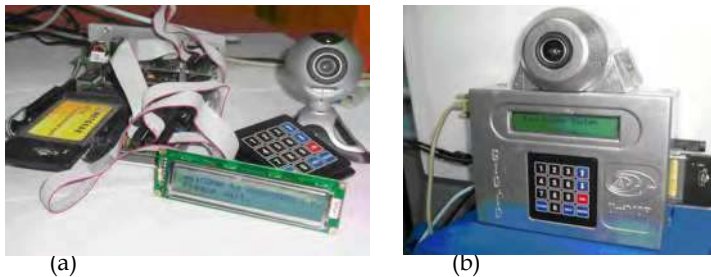(a)   (b)

Fig. 1. a) *Face Reader* components; b) Physical Model for *Face Reader*

The face image database and biometric identification software are embedded in the SBC; thus the recognition and verification of face images for small database system is performed on the board itself. For a large database system, external high speed PC server is used as a database to store face images and biometric identification software. The high speed PC server receives images and user ID send through network protocol (TCP/IP) and interface by *Face Reader*. Results from biometric identification is sent through a network to the *Face Reader* to be displayed through LCD panel. In order to accomplish the task (recognition and verification) by using the biometric identification algorithm, high speed processor PC Server based on GNU/Linux is chosen.

9.1 TS5500 SBC

The compatibility of an embedded system refers to the element of the processor, memory, I/O maps and BIOS. Memory for this model is dependant on the capacity of compact flash used within range from 32 MB – 1GB. SBC model from Technologic Systems, Inc is boot from IDE compact flash, DiskOnChip or On-board flash drive. TS5500 SBC (Figure 2) are compatible with several embedded OS with x86-based operating system. They are TSLinux, DOS, Windows CE, Net BSD, MicroCommander, SMX, QNX, Phar Lap, MicroC/OS II and eRTOS. The most popular OS used with this x86 models are DOS and TSLinux. This model has three COM ports and COM 2 is used to be a monitor for SBC using null modem. A serial

port is connected from the board at COM 2 to serial port at computer localhost. To enable the console at localhost to function well, a minicom should be installed first. Linux has provided a package that contains minicom. A default baud rate should be changed from 9600 to 115200. At the same time, the correct serial port that has been connected from localhost must be set at minicom configuration. The board is equipped with TSLinux 3.07a that has been preinstalled by Technologic Systems, Inc company before shipping. TS Linux is one type of embedded OS that is created by Technologic Systems, Inc and has many similarities like normal Linux features especially in filesystem but in small size.

Fig. 2. TS5500 SBC

Network support is one important feature for latest SBC technology. TS5500 has one RJ45 port and support standard network by using Telnet and file transfer protocol (FTP). But it does not support Secure Shell (SSH) function. Furthermore, the Secure CoPy (SCP) is allowed by this model by activating the dropbear function provided by TS Linux. The network point provides two LEDs to represent an active connection, and active data flow. On the other hand, PCMCIA is also reliable using 802.11 standard for wireless network. 5 Volt voltages are provided by and external power supply adapter connected to a standard 240V AC outlet. At the back of the unit is a reset button for reseting the unit to the factory defaults setting. The board comes with an AMD Elan 520 (equivalent to an Intel x86) processor that runs at 133MHz as well as 64 MB of RAM, a 2 MB flash disk, a Disk On Chip socket, and a PC/104 bus. It also has a Type 1 Compact Flash card reader, USB, PCMCIA a 10/100Base-T Ethernet interface, 40 Digital I/O lines and an alphanumeric LCD interface. The board requires 5V DC power at 800mA.

9.2 Development of Webcam Device Driver
The USB Webcam plays an important role in this project. The webcam is mounted on top of the *Face Reader* and attached to the SBC. This webcam is used for capturing face image which is then pre-processed and extracted for face facial recognition. Selecting the appropriate USB Webcam to integrate with the board requires a lot of testing and configuration. Most of the USB webcam is manufactured to comply with the Window environment; this means the manufacturer does not provide any supported driver for this webcam to work in Linux operating system. In this project, a few USB webcam models (Table 4) which are available in the market are chosen and tested for their compatibility with Linux operating system. This is done by utilizing the availability of open source and driver in Linux community. Results show that only two webcam can be configured and integrated

with Linux operating system. Besides the availability of the Linux device driver for specific webcam model, Linux kernels version is also an important issue in configuring and integrating the webcam with Linux operating system.

	Webcam Model	Linux Device Driver (V4L)
1	Creative Webcam Instant	Not Available
2	Logitech QuickCam Communicate STX/USB	spca500-20050101
3	Logitech QuickCam Express- USB	Not Available
4	Logitech QuickCam Chat	Not Available
5	Philip CU 2006	Not Available
6	Logitech QuickCam Pro 4000	pwc-8.8 & usb-pwcx-8.2.2

Table 4. Webcam Model and Linux V4L Device Driver

The integration and configuration of the webcam with TS Linux OS includes the testing which is done for Logitech QuickCam Communicate STX/USB and Logitech QuickCam Pro 4000 webcams. This testing is divided into a few features i.e. webcam workability in different kernel versions, image capturing and image readability. The testing is to find a suitable webcam that can be integrated with TS 5500 embedded PC. Besides TS Linux (kernel 2.4.23-25.ts), testing is also done in different platforms i.e. RedHat 8.1 (kernel 2.4.18-14), RedHat 7.3 (kernel 2.4.18-3), RedHat 7.3 (kernel 2.4.20) and RedHat 7.3 (kernel 2.4.23-25.ts). The selection of the USB Webcam Logitech QuickCam Pro 4000 for this project is based on results of the testing as shown in Table 5 and Table 6.

USB Web Camera	Logitech QuickCam Communicate STX		
Driver	spca500-20050101		
Features	Camera detected	Image capture	Image readability
RedHat8.1(kernel 2.4.18-14)	yes	Yes	yes
RedHat7.3(kernel 2.4.18-3)	-	-	-
RedHat7.3(kernel 2.4.20)	-	-	-
RedHat 7.3 (kernel 2.4.23-25.ts)	no	No	no
TS Linux (kernel 2.4.23-25.ts)	no	No	no

Table 5. Logitech QuickCam Communicate STX Webcam Configured with Different Linux Kernel

The TS 5500 SBC is installed with TS Linux version 3.07a with 2.4.23-25.ts kernel. Logitech Quickcam Pro 4000 web camera driver works in fresh kernel 2.4.20 and works well with the Philips web camera drivers modules i.e. PWC core modules (pwc-8.8.tar.gz) and PWCX decompressor modules (pwcx-8.2.2.tar.gz).

USB Web Camera	Logitech Quickcam Pro 4000		
Driver	pwc-8.8 & usb-pwcx-8.2.2		
Features	Camera detected	Image capture	Image readability
RedHat8.1 (kernel 2.4.18-14)	no	No	no
RedHat7.3 (kernel 2.4.18-3)	no	No	no
RedHat7.3 (kernel 2.4.20)	yes	Yes	yes
RedHat 7.3 (kernel 2.4.23-25.ts)	yes	Yes	yes
TS Linux (kernel 2.4.23-25.ts)	yes	Yes	yes

Table 6. Logitech QuickCam Pro 4000 Webcam Configured with Different Linux Kernel

9.3 Development of LCD Panel and Keypad Device Driver

Technologic Systems provide the documentation to integrate the SBC with Lumex LCD Panel by using LCDproc i.e. Linux LCD display driver. LCDproc is a piece of software that displays real-time system information from Linux box on a LCD. The server supports several serial devices and some devices connected to the LPT port. Various clients are available that display things like CPU load, system load, memory usage, uptime, and a lot more. LCDproc is consists of two parts: LCDd and user clients. LCDd is the deamon that interacts with the LCD display. User clients tell the deamon what to display on the LCD display. LCDd must be installed and running on the 32 bit embedded PC. The client need not install on the embedded PC or host itself because LCDd listens on a network interface.

The TS 5500 SBC support a matrix keypad provided by Technologic Systems. The matrix keypad code uses 8 digital I/O lines as row outputs and column inputs to scan a 4x4 keypad. The square wave output function of the real time clock is used as a pacing clock to generate interrupts on IRQ8. The interrupt service routine calls the keypad scans routine and provides debounce timing. The kernel module for the matrix keypad is ts_keypad.o, and exports the device node /dev/SBC/TSKeypad. When the device node is opened, the driver begins its keypad scan routine, storing key-presses into a buffer. When the device node is read from, if any key-presses were made, they a sent back to the user.

10. Software Development

BIOI²D software is developed on embedded GNU/Linux OS and SBC. The SBC is pre-installed with TS-Linux kernel 2.24.23. The selection of GNU/Linux OS is to utilize the availability of open source resources such as GCC compiler, libraries, kernels and drivers in developing and implementing this system. The *Face Reader* is designed to operate in real-time mode, which requires the face identification and recognition software to identify a person face, and pre-process the image. BIOI²D software system can operate in two (2) modes i.e. stand-alone and network mode. In stand-alone mode, all of the process in done on the *Face Reader* itself, while in network mode, server perform the biometric identification process using face image send by *Face Reader* through TCP/IP network protocol. BIOI²D software design is structured in five (5) modules i.e. User Interface, Image Acquisition, Image Preprocessing, Network and Biometric Identification. All software relies on an embedded version of the Linux OS and tailored with GNU C utilities. User interface module

incorporates the development of a system control program and the integration of LCD panel and matrix keypad to the SBC. The development of image acquisition module or the camera module involves the usage of open source Video4Linux (V4L) Application Programming Interface (API). Image is captured using YUV420P format and before the image is saved it is converted into greyscale format because image preprocessing algorithm and face recognition algorithm is functions only in greyscale format.

In image preprocessing module, image preprocessing algorithms are used to perform initial processing that performs primary data reduction and analysis task easier. The objective of image preprocessing module is to preprocess face image by converting the image to grayscale, removing the image background, improving image appearance and scale down the image. The output image is use by face recognition software for recognition purposes. Network mode involves client-server applications. Network module is developed by using socket API, which enable the communication between processes over the Internet, within the LAN or on single PC. The *sendfile* system call provides an efficient mechanism for copying data from one PC to another. In this mode, the client program sends the face image file and user ID variable to the server. The server performs face recognition and sending back the result to the client. Biometric identification module incorporates face recognition algorithm based on iris detection for identification and verification. Recognition system is based on template matching and the process is performed one-to-one matching (direct matching) by using user ID.

10.1 Image Pre-processing

The Face Reader is design to operate in real-time mode, which requires the face identification and recognition software to identify the person face from the captured image. The preprocessing algorithms are used to perform initial processing that makes the primary data reduction and analysis task easier. This include operations related to extracting regions of interest, performing basic algebraic operations on images, enhancing specific image features, and reducing data in both resolution and brightness. Face image is captured in color format and the image is with complex background. The objective of image preprocessing module is to preprocess face image by converting the image to grayscale, removing the image background, improving image appearance and scale down the image. The image preprocessing technique performs are the color space conversion, motion analysis technique, grayscale modification and image scaling. The output image is use by face recognition software for recognition purposes. Color space conversion is the process of changing one type of color-encoded signal into another. Face image is captured in YUV420P format. The conversion process implemented is converting from YUV420P to RGB and from RGB to Grayscale. In first conversion, source data is converted from the YUV420P color space to the RGB color model. A YUV image is then converted to a gamma-corrected RGB image. This function will returns the R, G, B component values that are equivalent to the color represented by the YUV420P values. . For RGB to grayscale conversion, it is clear that black grayscale is equal in value only to the RGB grayscale. The conversion formulas are:

$$R = Y + 1.593*(V-128) \qquad\qquad (3)$$
$$G = Y - 0.390*(U-128) - 0.515*(V-128) \qquad\qquad (4)$$
$$B = Y + 2.015*(U-128) \qquad\qquad (5)$$

The formula for conversion from RGB to Grayscale is as below:

Grayscale intensity = 0.2990R + 0.5870G + 0.1140B   (6)

Motion analysis is connected with real-time analysis. It is used to obtain comprehensive information about moving and static objects present in the scene. The input to a motion analysis system is a temporal image sequence, with a corresponding increase in the amount of processed data. For motion detection, it registers any detected motion and usually by using a single static camera. The software perform four steps to detect and analyze movement in image sequence i.e. frame differencing, threshold, noise removal and morphological filtering. Frame differencing operation is used to find the difference between the current image and the background image. If the difference between the pixel values is greater than constant value fixed, the movement has been significant and the pixel is set to black otherwise if the change is less than this threshold, the pixel is set to white. This image now indicates if something has moved and where the movement is located. To remove the noise, the program scans the threshold image with a 3 x 3 frame and removes all black pixels which are isolated in a white area. If the center pixel of the 3 x 3 frame is black and less than three of the pixels in the frame are black, the algorithm remove the black center pixel, otherwise the pixel remains black.

Morphological filtering simplifies a segmented image to facilitate the search for object of interest by smoothing out object outlines, filling small holes, eliminating small projections, and using other similar techniques. The two principal morphological operations are dilation and erosion. Dilation allows object to expand, thus potentially filling in small holes and connecting disjoint objects. Erosion shrinks objects by etching away (eroding) their boundaries. These operations can be customized for an application by the proper selection of the structuring element, which determines exactly how the objects will be dilated or eroded. Grayscale modification methods are a point operations and function by changing the pixel's (gray level) values by a mapping equation. The mapping equation is typically linear and maps the original gray level values to other specified values. The gray level histogram of an image is the distribution of the gray levels in an image. Histogram equalization is a popular technique for improving the appearance of a poor image where the histogram of the resultant image is as flat as possible. The theoretical basis for histogram equalization involves probability theory where the histogram is treated as the probability distribution of the gray levels. This is reasonable, since the histogram is the distribution of gray levels for particular image.

Image scaling is the process of changing the size of a digital image. These involve either increase or reduction of image size by a fixed ratio. Two properties of image scaling are the contraction ratio is the same everywhere, although pixel distances are not preserved and the pixel lattice of the target image is a subset or superset the original image for an integer scaling ratio. The first property allows the use of spatially invariant filter, and the second property avoids the need for resampling when the scaling ratio is an integer. Scaling down process increases the incidence of high frequencies and causes several pixels to collapse into one. In order to minimize aliasing problems in the target image, Box filter (the smoothing filter) is applied.

11. Results and Discussion

BIOI²D feasibility is tested by setting-up a test environment in the laboratory. The experimental setup consists of a *Face Reader* prototype and a PC as a server. The *Face Reader* prototype and the server are connected to the LAN switch through wired Ethernet TCP/IP network cable. The evaluation is carried out for both stand-alone and network operation modes. In stand-alone mode, face recognition process is executed on the board (SBC), and for network-mode the face recognition process is implemented using external program run on a server using TCP/IP network through a 5 port switch. The experiments are made by using 100 face images of 10 different persons for the evaluation of the accuracy of biometric identification. This includes setting-up face template and database images. The person's face image is captured and pre-processed before biometric identification process is performed. Output image file from image preprocessing process and user ID are transmitted through the LAN to the server by utilizing socket programming interface. At the server, the image file and user ID's are used for biometric identification process and the result is sending back to the *Face Reader* through a LAN (whether the face is match with the information in the database). The *Face Reader* displays the result (such as "Authentication successful") though LCD panel. The delay measurement for both operation modes is counted from the beginning of image acquisition process until the ends, when the identification result is displayed.

Specification	Desktop PC	SBC
Processor Type	Intel(R) Core (TM) 2 CPU	AMD Elan 520
Processor Speed	1.8 GHz	133 MHz
RAM Size	1 GB	64 MB
Operating System	Mandriva 2006	TS-Linux

Table 7. Comparison desktop PC and SBC Specifications

BIOI²D hardware performance evaluation is performed by measuring the processing time for the overall BIOI²D operation, image pre-processing process and biometric identification process. For BIOI²D operation, the comparison is made to measure the performance for stand-alone and network mode. In image pre-processing and biometric identification process the performance of the 133 MHz SBC is compared to a desktop PC with Intel(R) Core (TM) 2 CPU running at 1.83 GHz (desktop). Table 7 shows the details of a desktop PC and SBC specifications. Overall BIOI²D processing time is measured starting from camera initializing until biometric identification results display on LCD Panel for both stand-alone and network modes. The processes include image capturing, user identification number input, image preprocessing, biometric identification and result display. Network mode requires additional process which involves the use of socket interfaces for network connectivity between *Face Reader* and a server for the face image, user identification number and result transfers. Table 8 shows the comparison for overall BIOI2D operations for both modes. Results show that the network mode is faster than stand-alone mode. The overall stand-alone mode processing speed is only 62.4% of a network mode.

BIOI²D Operation	Overall Processing (s)
Stand-alone	190.53
Network	118.95

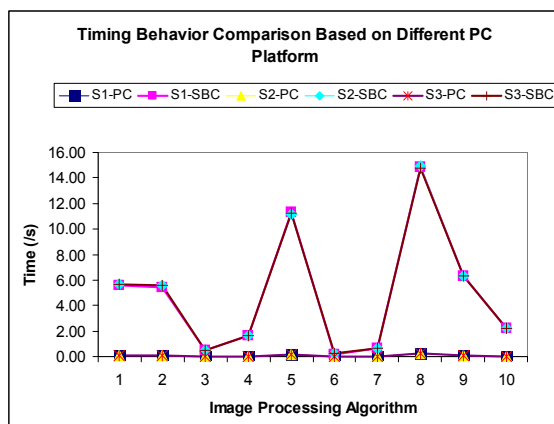Table 8. Comparison of overall BIOI²D operations execution time

Fig. 3. Timing Behavior Comparison Based on Different PC Platform

The processing time for the image preprocessing process is evaluated by executing the algorithm by using 3 different image samples. The three (3) image samples differ in their background and local contrast. The speed for the *Face Reader* (SBC) is only 1.52 % of a desktop platform for image pre-processing process as shown in Figure 3 which illustrates timing behaviour comparison based on different PC platform. The pattern demonstrates that the processing time for image preprocessing algorithm is almost the same even though the complexity of the background image and   the image contrast differs. Biometric identification performance evaluation is to measure the executing time for biometric identification process by executing the algorithm on both SBC and desktop platforms. Table 9 shows the result of the execution time which shows that processing is faster on desktop compare to SBC. For SBC platform the processing time are 0.01 operations per seconds while for desktop the execution time are 1.11 operations per second. The speed of SBC for biometric identification algorithm is slower than desktop that is only 0.9 % of a desktop speed.

As shown in Table 7, the SBC and server differ in processor type, processing speed, RAM size and OS. The hardware specification for the server is much faster than SBC. In stand-alone mode, all processes are execute on the prototype model itself, thus the processing time is dependence solely on the SBC hardware. For network mode, biometric identification process is performed on the server. This mean the processing time for biometric identification process on the server is much faster than in the *Face Reader*. The CPU processing speed and short term memory (RAM) are the key factors of determining the performance of the overall processing time for the system. The performance for network mode also depends on TCP/IP network performance such as speed, bandwidth, throughput and latency. Regardless of slower performance of SBC on the hardware performance, this functionality can still be a part of a successful embedded biometric identification system based on iris detection and the results also shows that the software is compatible for both platforms i.e. SBC and desktop.

Platform	Biometric Identification process (seconds)	Operations per second (ops)	% differences (SBC)
SBC	71.78 s	0.01 ops	0.9 %
Desktop	0.9 s	1.11 ops	

Table 9. Comparison of biometric identification process

The *Face Reader* is designed to operate in real-time mode, which requires the face identification and recognition software to identify the person face from the captured image. Image preprocessing processes are used to perform initial processing for the captured image. Face image is captured in YUV420P color format with complex background. Color space conversion technique is used for the conversion image colour format from, YUV420P to Greyscale format. Motion analysis technique is to locate the face and remove the background of the image. Histogram equalization technique improved image appearance by increase image local contrast and sharpness. The image is scaled down by using image scaling technique. Figure 4 shows the input, background and preprocessed output face image.

| Input Image | Background | Output Image |
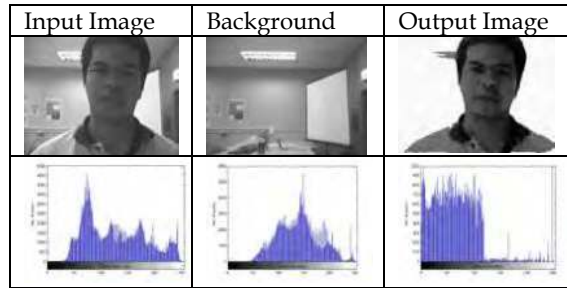
Fig. 4. Input, background and preprocessed output face image.

Figure 5 shows examples iris identification result of the face images which the biometric identification algorithm gave successful results and also biometric identification algorithm failed to detect the positions of the irises. All errors occurred in biometric identification were due to the biometric identification algorithm failures of iris detection. Figure 6, shows the matching percentages for difference image samples. The matching rate is the comparison between the same person in the database and image captured from the *Face Reader*. If the test successfully identifies the correct person, the result is success, vice versa it is fail. Table 10 shows the identification result for biometric identification software. The results show that the successful rate for the biometric identification system is 73% and percentage of matching that is reliable and should be used as a threshold for system implementation is 98%.

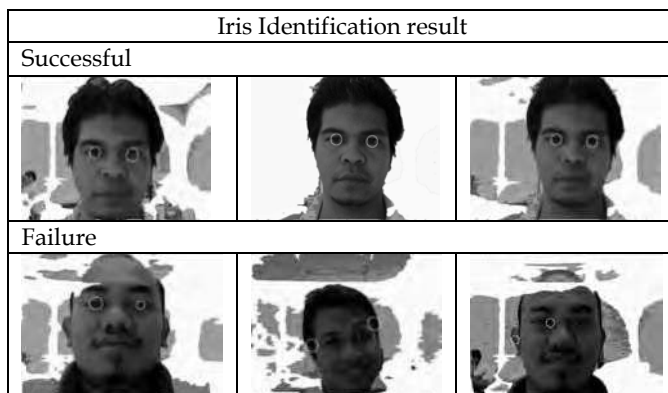| Iris Identification result |
| Successful |

| Failure |

Fig. 5. Sample of the face images which the biometric identification algorithm gave successful and failure iris identification result.
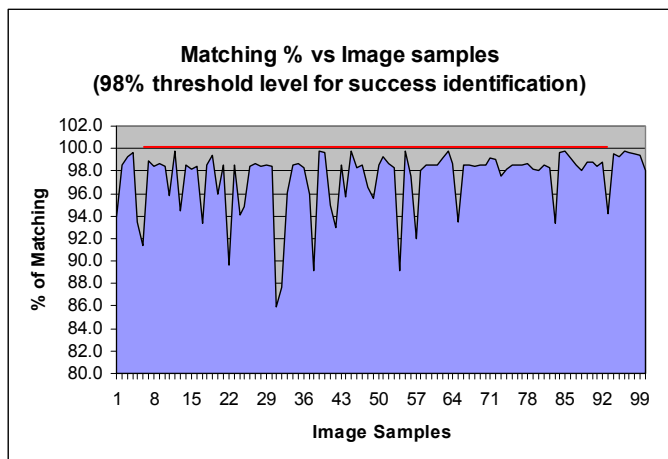

Fig. 6. Results for face recognition algorithm, graph show matching percentage vs. image samples, if percentage is 98% and above, identification is successful.

Total number of images	Number of successful images	Number of failed images	Success rate (%)
100	73	27	73

Table 10. Identification Results for Different Image Samples

12. Conclusion and Future Work

This chapter describes the design and implementation of an Embedded System for Biometric Identification from hardware and software perspectives. The first part of the chapter describes the idea of biometric identification. This includes the definition of

biometric and general biometric system design. It also emphasize on the number biometric characteristics exist and are in use in various applications. A brief comparison of various biometric techniques based on seven factors and the applicability of a specific biometric technique depends on the requirements of the application domain. This part also highlights on face recognition as one of the most successful applications of image analysis and understanding with numerous practical applications. This chapter also discusses on the emerged of embedded systems and various projects with embedded hardware platforms. It also highlights several projects which focus on the integration of image acquisition and processing in single embedded device.

Hardware implementations focus on design, hardware integration and configuration in developing an embedded system for biometric identifications. It is based on single board computer (SBC) and utilizing GNU/Linux operating system (OS) which allows the use of open source resources such as libraries, kernels drivers and GNU C compiler in developing and implementing this system. This involves explaining the effort on finding compatible external devices I/O that can be integrate with the SBC, which include devices such as LCD Display, Matrix Keypad, Compact Flash Card, PCMCIA Wireless Network Card and USB Web camera. This embedded system is designed to operate in real-time mode to execute the following tasks: face (image) capture, pre-processing and matching with database using predetermine user identification number to reduce the processing tasks. Software design is structured in five modules namely as User Interface, Image Acquisition, Image Pre-processing, Network and Biometric Identification.

This chapter also present hardware and software design for BIOI²D based on SBC and GNU/Linux. The hardware includes single board computer, USB webcam, LCD display panel and matrix keypad. The software components are based on GNU C programming language and Linux OS. The experimental results show that the algorithm is capable of performing the image pre-processing task. The system able to capture face image, execute image pre-processing and perform biometric identification based on iris detection. Future work includes reducing the face reader size, improving face recognition algorithm and increasing processing speed. This chapter demonstrate that embedded processing technology, in particular the x86 processor TS-5500 SBC, has been developed well enough to make it useable for implementing a computer vision system adequate for embedded biometric identification system.

13. Acknowledgement

14. References

Aaraj, N., Ravi, S., Raghunathan, A., and Jha, N. K., (2006) Architectures for efficient face authentication in embedded systems, *Proceedings of the conference on Design, automation and test in Europe: Designers' forum*, Munich, Germany, EDAA, p. 1-6,

Abreu, B., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., unes, L., Orri, J., Trigueiros, M.J. and Violante, A.,"Video-based multi-agent traffic surveillance system," in Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE, 2000, pp. 457 – 462.

Al-Ali, A., Al-Rousan, M., and Al-Shaikh, M. (2003) Embedded system-based mobile patient monitoring device. In *Computer-Based Medical Systems, 2003. Proceedings. 16th IEEE Symposium*, pages 355– 360, 2.3.7

Baber, C. and Baumann, K., (2002) Embedded human computer interaction. *Applied Ergonomics*, 33(3):273– 287, 2.3.7

Badlishah, R. Ahmad, Wan Muhamad Azmi Mamat, Ahmad Nasir Che Rosli, (2006a) System Application Development : Single Board Computer (SBC) & Gnu/Linux for Robotic Application, *Proceedings of the International Conference on Underwater System Technology : Theory and Applications 2006 (USYS'06)*, USM, Penang.

Badlishah, R. Ahmad, Wan Muhamad Azmi Mamat, Ahmad Nasir Che Rosli, Suhizaz Sudin, (2006b) Embedded Linux in KUKUM: Towards High End Embedded System Product Design, *Malaysian Technical Universities Conference on Engineering and Technology 2006 (MUCET 2006)*, KUiTHHO

Bramberger, M., Doblander, A., Maier, A., Rinner, B. and Schwabach, H. (2006)"Distributed embedded smart camera for surveillance applications", Computer, vol. 39, no. 2, pp. 68-75.

Blackburn, D., Bone, M., and Phillips, P. J., (2001) Face Recognition vendor test 2000. Tech. rep. 2001. http://www.frvt.org.

Cao Z.Y., Ji, Z.Z. and Hu, M.Z. "An image sensor node for wireless sensor networks," in Proc. International Conference on Information Technology: Coding and Computing (ITCC 2005), Apr. 2005, vol. 2, pp. 740-745.

Chellappa, R., Wilson, C.L. and Sirohey, C., (1995) "Humain and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705-740.

Comerford, R., (1998) Pocket computers ignite OS battle. *IEEE Spectrum*, May. 2.2

Dan, H., (2003) An autonomous mobile robot development platform for teaching a graduate level mechatronics course. *33rd ASEE/IEEE Frontiers in Education Conference.*

Deb, S., Ghoshal, S., Malepati, V. and Kleinman, D., (2000) Tele-diagnosis: remote monitoring of large-scale systems. In *Aerospace Conference Proceedings, 2000 IEEE*, volume 6, pages 31–42, 2.3.7

Downes, I., Baghaei R.L. and Aghajan H., "Development of a mote for wireless image sensor networks," in Proc. COGnitive Systems with Interactive Sensors (COGIS 2006), Mar. 2006.

Dufey, J, Jost, B., Neufeld, N. and Zuin, M. (2000) Event building in an intelligent network interface card for the lhcb readout network. In *Nuclear Science Symposium Conference Record, 2000 IEEE*, volume 3, pages 26/50 –26/53, 2.3.7

Fleck S. and Straßer, W., "Adaptive Probabilistic Tracking Embedded in a Smart Camera," in IEEE Embedded Computer Vision Workshop (ECVW) in conjunction with IEEE CVPR 2005, 2005, pp. 134 – 134.

Fleck S., Busch, F., Biber, P. and Straßer, W. "3D Surveillance – A Distributed Network of Smart Cameras for Real-Time Tracking and its Visualization in 3D," in Computer Vision and Pattern Recognition Workshop, 2006 Conference on, Jun. 2006, pp. 118 – 118.

Fleck S., Busch, F., Biber, P. and Straßer, W. "Adaptive Probalistic Tracking Embedded in Smart Cameras for Distributed Surveillance in a 3D model", EURASIP Journal on Embedded Systems, 2007.

Fryer, R., (1998) Low and non-intrusive software instrumentation: a survey of requirements and methods. *In Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, volume 1, pages C22/1 –C22/8, 2.3.7

Garcia, R.L., Carlos Alberola-L'opez, Otman Aghzout and Juan Ruiz-Alzola, (2003) Biometric identification systems, Signal Processing, Vol. 83, Issue 12, pp. 2539 – 2557.

Galton, F., (1888) "Personal identification and description," *In Nature*, pp. 173-177, June 21.

Gori, L., Tommasini, R., Cautero, G., Giuressi, D., Barnaba, M., Accardo, A., Carrato, S. and Paolucci, G. (1999) An embedded control and acquisition system for multichannel detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 431(1-2):338–346, 2.3.7

Hengstler, S. and Aghajan, H.,"A smart camera mote architecture for distributed intelligent surveillance" in International Workshop on Distributed Smart Cameras (DSC-06) in conjunction with The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), 2006.

Jain, A.K. Bolle, R. and Pankanti, S. (eds.), (1998) *Biometrics, Personal Identification in Networked Society*, Kluwer Academic Publishers, Norwell, MA.

Jain, A.K., Hong, L. and Pankanti, S. (eds.), (2000) *Biometric Identification*, Communications of the ACM, February, ACM Press, New York USA, Vol. 43, Issue 2, pp 90-98.

Jain, A.K, Arun Ross and Salil Prabhakar, (2004) An Introduction to Biometric Identification, in IEEE Transaction on Circuits and Systems for Video Technology, Special Issue on Image and Video-Based Biometrics, Vol. 14, No. 1.

Kleihorst, P., Schueler, B., Danilin, A. and Heijligers, M., "Smart camera mote with high performance vision system" in International Workshop on Distributed Smart Cameras (DSC-06) in conjunction with The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), 2006.

Kroll, M., Schütze, B., Geisbe, T., Lipinski, H.G., Grönemeyer, D.H.W. and Filler, T.J. (2003) Embedded systems for signing medical images using the dicom standard,. In *International Congress Series*, volume 1256, pages 849–854, 2.3.7

Lamberti, F and Demartini, C. (2003) Low-cost home monitoring using a java-based embedded computer. In *Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference*, pages 342–345, 2.3.7.

LinuxDevices.com., (n.d.) A Linux-oriented Intro to Embeddable Single Board Computers. Citing Internet Sources. Retrieved on 2007-08-21 from URL http://www.linuxdevices.com/articles/AT6449817972.html.

Messer, K., Matas, J., Kittler, J., Luettin, J., and Maitre, G., (1999) XM2VTSDB: The Extended M2VTS Database. In *Proceedings, International Conference on Audio and Video-Based Person Authentication*. pp. 72-77.

Oliveira, F.D.R., Chalimbaund, P., Berry, F., Serot, J. and Marmoiton, F. "Embedded early vision systems: implementation proposal and hardware architecture," in Proc. COGnitive Systems with Interactive Sensors (COGIS 2006), Mar. 2006.

Perera, A., Papamichail, N. , Bârsan, N., Weimar, U., Marco, S., (2003) On-line Event Detection by Recursive Dynamic Principal Component Analysis and Gas Sensor Arrays under drift conditions, *In Proceedings of the 2nd IEEE international conference on sensors*, Toronto, Canada.

Phillips, P. J., Wechsler, H., Rizvi, S., and Rauss, P., (1998) The FERET database and evaluation procedure for face-recognition algorithms. Image Vis. Comput. 16, pp. 295-306.

Phillips, P. J., Moon, H., Rizvi, S., and Rauss, P., (2000) The FERET evaluation methodology for face recognition algorithms, IEEE Trans. Patt. Anal. Mach. Intell. 22.

Phillips, P. J., Grother, P. J. , Micheals, R. J., Blackburn, D. M., Tabassi, E., and Bone, J. M., (2003) Face Recognition vendor test 2002: Evaluation report. NISTIR 6965, http://www.frvt.org.

Quaritsch, M., Kreuzthaler, M., Rinner, B., Strobl, B.,"Decentralized Object Tracking in a Network of Embedded Smart Cameras" in International Workshop on Distributed Smart Cameras (DSC-06) in conjunction with The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), 2006, pp. 99 - 104

Rahimi M, Baer, R., Iroezi, O.I., Garcia, J.C., Warrior, J., Estrin, D. and Srivastava, M. "Cyclops: In situ image sensing and interpretation in wireless sensor networks," in Proc. 3rn International Conference on Embedded Networked Sensor Systems (SenSys '2005), Nov 2005, pp. 192-204.

Ramamritham, K. and Arya, K. (2003) System support for embedded applications. In *Proceedings. 16th International Conference on VLSI Design, 2003*, pages 22–26, 2.3.7

Ratha, N.K., Senior, A., Bolle, R.M., (2001) Tutorial on automated biometrics, in *Proceedings of International Conference on Advances in Pattern Recognition*, Rio de Janeiro, Brazil.

Remagnino, P., Orwell, J., Greenhill, D., Jones, G.A. and Marchesotti, L., Multimedia Video Based Surveillance Systems: Requirements, Issues and Solutions. Kluwer Academic Publishers, 2001, ch. An Agent Society for Scene Interpretation, pp. 108 – 117.

Richard A. Sevenich, (2004), *An Introduction to Embedded Linux Development: Part 1*, Citing Internet Source, Retrieved March 20, 2006, URL: http://www.linuxdevices.com/.

Rizvi, S. A., Phillips, P. J., and Moon, H., (1998) A verification protocol and statistical performance analysis for face recognition algorithms. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*. Pp. 833-838.

Robert A. Burckle, (n.d.) The evolution of Single Board Computers. Citing Internet Sources. Retrieved on 2007-08-21 from URL http://www.winsystems.com

Rowe, A., Rosenberg, C. and Nourbakhsh, I. "A second generation low cost embedded color vision system," in IEEE Embedded Computer Vision Workshop (ECVW) in conjunction with IEEE CVPR 2005, vol. 3, 2005, pp. 136 – 136.

Stepner, D., Nagarajan Rajan, David Hui , (1999) Embedded Application Design Using a Real-Time OS, *Proceedings of the 36th ACM/IEEE conference on Design automation DAC '99*.

Tan, T., Raghunathan, A. and Jha, N. (2003) A simulation framework for energy-consumption analysis of os-driven embedded applications. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, pages 1284– 1294, 2.3.7

Tolba, A. S., El-Baz, A. H. and El-Harby, A. A, (2005) Face Recognition: A Literature Review, International Journal of Signal Processing, Vol. 2, No. 2, pp 88-103.

Velipasalar, S., Schlessman, J., Chen, C.-Y., Wolf, W. and Singh J. P., "SCCS: A Scalable Clustered Camera System for Multiple Object Tracking Communicating via Message Passing Interface," in Multimedia and Expo, 2006. ICME 2006. IEEE International Conference on, 2006.

Wiencke, L., (2006) TS5500 Single Board Computer, Citing Internet sources URL http://www.physics.utah.edu/~wiencke/pc104html/5500.html

Williams, A., Xie, D., Ou, S., Grupen, R., Hanson, A. and Riseman, E. "Distributed smart cameras for aging in place" in International Workshop on Distributed Smart Cameras (DSC-06) in conjunction with The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), 2006.

Wolf, W., Ozer, B. and Lv, T. "Smart cameras as embedded systems," Computer, vol.35. no 9, pp. 48-53, 2002.

Wood, S., (2006) Using many PC/104 serial ports. Citing Internet sources URL http://www.jlab.org/~Esaw/pc104/

Zaho, W., (1999) "*Robust image based 3D face recognition*," Ph.D. Thesis, Maryland University.

Zhao, W., Rama Chellappa, P.J. Jonathon Phillips, and Azriel Rosenfeld, (2003) Face Recognition: A Literature Survey, ACM Computing Survey, December Issue, pp. 399-458.
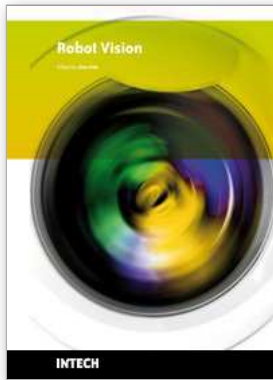
Robot Vision

Edited by Ales Ude

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ahmad Nasir Che Rosli (2010). Embedded System for Biometric Identification, Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, Available from: http://www.intechopen.com/books/robot-vision/embedded-system-for-biometric-identification

INTECH
open science | open minds