Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 41 (2012) 395 – 404

Engineering Procedia

www.elsevier.com/locate/procedia

International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)

Machine Vision for intelligent Semi-Autonomous Transport (MV-iSAT)

Kenneth Mapanga[a] and Veera Ragavan Sampath Kumar[a]*

[a]*School of Engineering, Monash University, Sunway Campus, Jalan Lagoon Selatan, 46150 Bandar Sunway, Selangor, Malaysia*

Abstract

The primary focus was to develop a vision-based system suitable for the navigation and mapping of an indoor, single-floor environment. Devices incorporating an iSAT system could be used as 'self-propelled' shopping carts in high-end retail stores or as automated luggage routing systems in airports. The primary design feature of this system is its Field Programmable Gate Array (FPGA) core, chosen for its strengths in parallelism and pipelining. Image processing has been successfully demonstrated in real-time using FPGA hardware. Remote feedback and monitoring was broadcasted to a host computer via a local area network. Deadlines as short as 40ns have been met by a custom built memory-based arbitration scheme. It is hoped that the iSAT platform will provide the basis for future work on advanced FPGA-based machine-vision algorithms for mobile robotics.

Keywords: FPGA image processing, real-time data path, machine vision, mobile robot platform.

1. Introduction

Vision-based mapping is a contemporary and growing research area. The use of Field Programmable Gate Arrays (FPGAs) has been explored in this research so as to ascertain the suitability of the technology to this particular application. Naturally, the attractiveness of FPGAs lie in their ability to perform both tasks and data operations in parallel. Consequently, data-path design was an integral part of this research.

According to Davies [1], *machine vision* is the study of methods and techniques enabling the construction and use of artificial vision systems in practical applications. The growth of this technology is largely hinged on the falling cost of CMOS cameras and thus its positioning as a good value-for-money choice for acquiring sensory information about a machine's surroundings. The iSAT is based on a machine vision system for this reason.

The qualitative descriptor *intelligent* is difficult to describe. Schlkoff recognised this and expressed *intelligence* in two ways. Firstly, to an average person an intelligent machine behaves the way the ones in the movies do [2]. Secondly, or more formally it refers to representations, inference procedures and learning strategies [2]. The intelligent aspect of iSAT points towards its anticipated ability to solve a transport problem with limited human intervention. Intelligence begins with small steps such as a data-processing framework that can interpret the floor within a scene. This is the level of intelligence being demonstrated here.

Typical autonomous transport systems require substantial a priori knowledge of their environment. This is true for most Automated Guided Vehicles (AGVs), as they require definite paths and demarcations to perform their duties. Conversely, iSAT could be applied to constantly changing scenarios. Real-time constraints have been addressed through the use of hardware-software co-design.

* Corresponding author. Tel: +60 3 5514 6326; Fax: +60 3 5514 6207.
E-mail address: veera.ragavan@eng.monash.edu.my

1877-7058 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2012.07.190

Real world problems demand *smart* transport devices. Whilst airport luggage systems appear automated above ground, behind the scenes a myriad of workers manually sort, transport and load up to 100 bags per minute [3]. Other novelty applications exist too. Retailers could introduce *self-pushing* supermarket trollies by incorporating iSAT. These trollies could follow their instructor and negotiate obstacles. Data could be collected about shopper habits, how they travel through the retail space and trolley theft could also be reduced as trolleys would promptly self-navigate back to base.

2. Literature Review

Researchers have individually pursued mapping for vision-based localisation [4], using high-end embedded smart-camera systems [5], real-time image processing using FPGAs [6], monocular obstacle detection [7] and visual Simultaneous Localisation and Mapping (vSLAM) [8] [9]. In order to make iSAT a commercial product, a combination of these technologies is likely to be required. In addition to academic information, product details from various suppliers and manufacturers were also analysed and are presented here.

2.1. Choice of Primary Sensor

Cameras supply a large quantity of data about the environment for a relatively low cost. What needs to be determined is the integration level. Although ready-made camera-capture devices already exist, these high level devices put significant performance and resource constraints on the rest of the system. Nonetheless, these CMOS or CCD cameras come pre-integrated with some form of on-board processing. The stalwart of this category of device is the Surveyor SRV-1 Blackfin Camera [10]. On a smaller scale the CMUcam series is a lower-cost alternative with an integrated CMOS sensor and ARM7 processor [11]. At the other extreme, a USB webcam could be used. For commercial applications, typically CMOS imagers are interfaced directly. Using a board level camera with a Field Programmable Gate Array (FPGA) has already been demonstrated [12]. Similarly some embedded processors [13] [14] [15] can interface directly to CMOS cameras. Board level CMOS cameras were therefore targeted for the iSAT machine vision system.

2.2. Image Processing Algorithms

Filtering, thresholding, edge detection, pattern analysis and transformations [1] are all common starting points for image processing. Significant work has been done on edge-detection methods involving Canny, Sobel and Robert's Cross techniques [16] on FPGA systems. Appearance-based obstacle detection is an interesting method [17] which seems to have lost favour with current research approaches. It has been selected as a benchmark for the hardware-based mobile robot vision system presented in this paper.

By assuming the floor is immediately in front of or under the robot, it was shown that it is possible to classify pixels as belonging to either the floor or an obstacle by using a colour histogram in the hue-saturation-value (HSV) colour space [17]. Good results were presented [17] and the aim of this work was to achieve comparable performance using an FPGA in an attempt to simplify the design and reduce the cost, size and power requirements of the machine vision system, with a view towards commercialisation.

This algorithm was successfully implemented during the course of this study. The future lies in performing more advanced algorithms on an FPGA. Examples could include implementing the Scale Invariant Feature Transform (SIFT) [18] or Speeded-Up Robust Features (SURF) [19]. These algorithms are geared towards the implementation of visual Simultaneous Localisation and Mapping (vSLAM) [20], which could form the core supervisor for an iSAT trolley.

2.3. Machine Vision System

FPGAs have been identified as the target device for the machine vision system. The reasons for this surround FPGAs' relatively low power requirements and 8x performance jump over a dual core 2.4GHz processor [19] when performing image processing on multiple sub-images at once [21]. Other suggestions for the central controller came from [22]. Apart from FPGAs [23], single-board computers (CPUs) [15] [24], graphics processors (GPUs) [25] and hybrid ARM-FGPA devices [26] [27] [28] were considered.

As shown in a study conducted in Florida [21] CPUs, even when arranged as multiple, cores with hyper-threading, suffer from having to sequentially process a very limited number of pixels at a time despite being able to operate in the gigahertz. GPU's at the other extreme, have a multitude of cores. However, communication between cores is limited, not to mention that GPU processing for purposes other than display output is still a very new area without a significant resource base to rely on [22]. The nVidia CUDA system will likely lead the way forward in GPU image processing [25]. Although this is beyond the scope of this work, what is clear is the trend to take real-time image processing into an accelerated multi-core

environment. Given the applicability and technology available, an FPGA controller is ideally positioned for a mobile robot vision system.

2.4. Mobile Robotics

Several kinematic models are available. Differential drive, omnidrive, Mecanum-drive and synchrodrive systems as well as Ackermann and Double Ackermann steering are all possible for a mobile robot [29]. With these models presented, it was decided that differential drive would be pursued because of the flexibility it allowed for the vehicle whilst still allowing it to bear heavy loads.

2.5. Literature Based Decision-Making

The component selection procedure involved researching devices available globally and resulted in the selection of the monochrome GameBoy™ camera [30] for preliminary work using a Freescale Smart Car Kit [31] and the C3188A Colour CMOS camera [32] for the final trolley. The DE2-115 [23] was selected as the MV-iSAT hardware-software development platform mainly due to familiarity with the FPGA design tools. As a result of specification, components for the propulsion system were also identified [33] [34].

The gap identified from the literature reviewed was the lack of academically available information on the construction of a real-time hardware based data path for image processing using FPGA design. Although demonstration intellectual property was typically available from major corporations, attempts to obtain descriptions or source code of possible data paths were futile. A sincere attempt has been made through this work to provide academic information on the design of a data-path for an entire FPGA-based machine vision system beginning with the camera input and ending with the robotic actuation. The design methodology and procedure follow.

3. Design Methodology

The mechatronics model was used to guide the design process. Kevin Craig's model describing mechatronics [35] is the de facto standard when pursuing synergistic design. He describes mechatronics as the "synergistic integration of physical systems, sensors, actuators, electronics, controls and computers through the design process enabling complex decision making" [35].

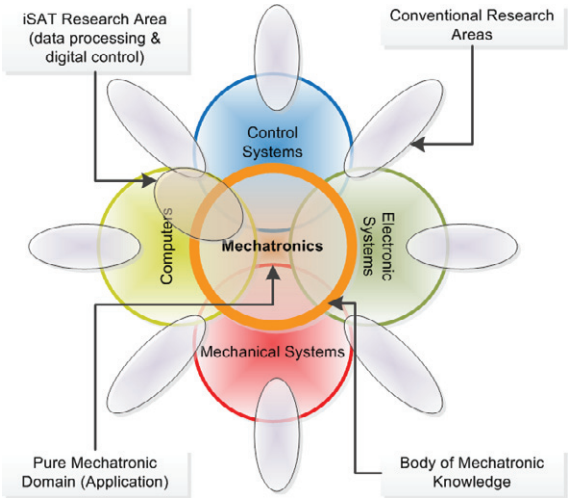

Fig. 1. Mechatronic design model applied to the development of MV-iSAT (adapted from the work of Kevin Craig [35]).

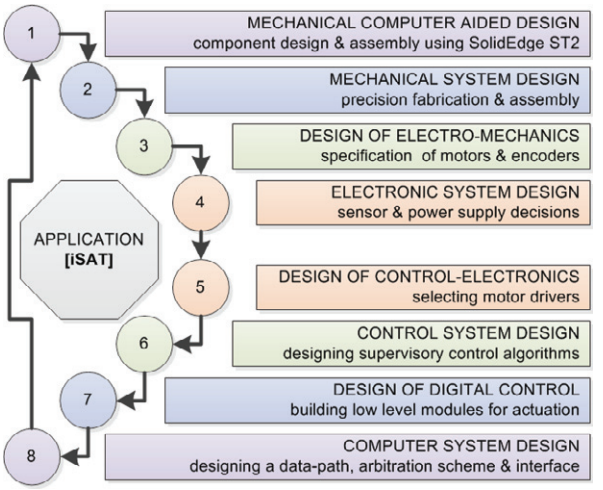

Fig. 2. Iterative development of MV-iSAT as a transport system (implementation of Kevin Craig's Mechatronic Design Model [35]).

Perhaps the only draw-back of this model is the absence of a discrete body of knowledge belonging to the mechatronic domain. It is hereby proposed to extend Craig's model to include application development at the centre of the mechatronic domain. This adaptation is illustrated in *Figure 1*.

By appropriately conducting product market research, system conceptual design and detailed design [36], mechatronic engineers can create synergies. They do this by carefully segmenting a problem across engineering domains. They follow up

by selecting the most appropriate components or modules. Finally they use iterative co-design and development to arrive at a more than proportional outcome by integrating a careful group of less complex techniques drawn from across domains which collectively solve the problem. This concept was the theoretical thrust behind the development of iSAT. The experimental investigation focusing on hardware-software co-design of a vision system is also highlighted in *Figure 1*.

4. Design Procedure

The team entered the Freescale Smart Car competition [31] in order to gain resources, guidance and experience towards testing a toy-model of iSAT. Subsequently a full-size iSAT trolley prototype was developed based on the model of a differential drive mobile robot. Following the mechatronic methodology by Craig [35] eight design aspects were co-designed iteratively as described in *Figure 2 above*.

4.1. Mechanical Computer Aided Design

Computer Aided Design (CAD) was a technique used to design and correct the prototype using 3D graphics before fabrication. Accurate dimensioning of purchased components enabled precision aluminium brackets, couplings and mountings to be designed. The isometric view of the chassis is shown in *Figure 3*.

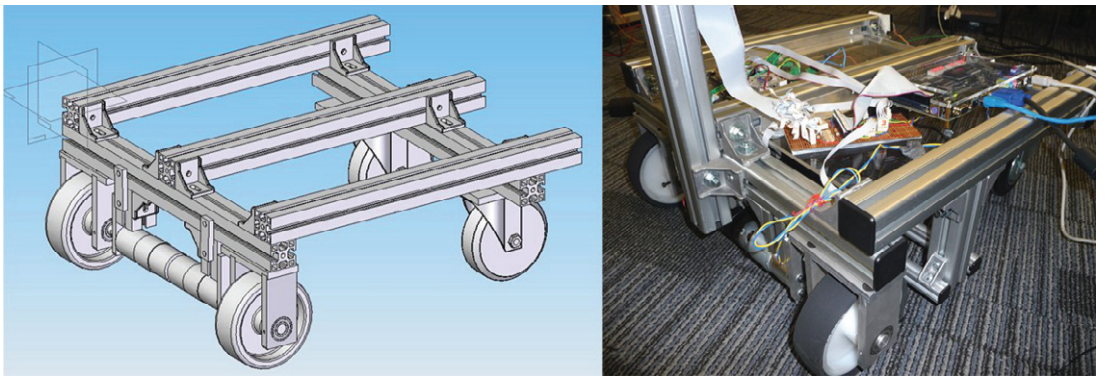

Fig. 3. Initial isometric view of the trolley in CAD followed by the resultant physical assembly after iterative co-design and development.

4.2. Mechanical System Design

The use of CAD resulted in well-fitting parts. This was especially necessary for the rear wheel housing and shaft as it was mounted rigidly by multiple members. The coupling was innovative as no standard couplings could bridge a 6mm motor shaft with a 15mm load-bearing drive-wheel shaft. The solution was a custom coupling made from interlocking a '+' shaped protrusion with its inverse being milled out to form a socket. A layer of silicone rubber was woven around the mating surfaces for cushioning. A Perspex sheet was used as the load-bearing surface.

4.3. Design of Electro-Mechanics

Considerations were made regarding the payload and the resulting motor torque, speed and control requirements whilst bearing cost in mind as well. Design calculations based on transporting 50kg at walking pace (1m/s) for a trolley with a nominal self-weight of 10kg resulted in the selection of 2 DC-geared motors with pre-assembled Hall-effect rotary sensors [33]. Each motor selected had a rated mechanical output power of 28W that maxes out at 70W. This provided a design safety factor of 2.

4.4. Electronic System Design

The primary sensor selected was the OV7620 1/3 inch CMOS camera chip [32] obtained as the C3188A [37] module. The 5V camera system was harmoniously interfaced with the 3.3V FPGA using current limiting resistors with zener diodes for over- and under- voltage protection.

The vehicle uptime was set at 2 hours. The current draw of all equipment was summed resulting in the selection of a 12V,

26Ah battery, which surpassed the 20Ah required. After the battery was procured, an undercarriage was designed to avail space beneath the load-bearing platform.

4.5. Design of Control Electronics

Control electronics were included in the design to connect the digital logic to the motion of the vehicle. Motor drivers were employed. Consideration points were the operating frequency, peak current ratings, efficiency and interface. The interface selected used two TTL-logic direction signals and a pulse-width-modulated speed signal. A suitable device was located [34]. These electronic components were mounted in a purpose-built cranny beneath the load-bearing board. Adequate ventilation was ensured.

4.6. Control System Design

The benchmark image processing algorithm was implemented in hardware to provide a proof-of-concept controller for the MV-iSAT prototype. This experimental investigation proved that image processing could be conducted in real-time using a system on chip data path. Other controllers are possible, including high level scene recognition [18] [19], mapping [9] and path planners [38].

4.7. Design of Digital Control

The digital control system was designed to be a hardware block for steering and propelling the vehicle appropriately. A pulse width modulation (PWM) component was designed for controlling each drive wheel's speed. A proportional-integral-derivative PID controller was directly overlaid for handling transient effects. By combining these two steps into one hardware module redundant calculations were removed.

4.8. Computer System Design

The vision processing system and the corresponding data path were the main custom-designed components of this experimental investigation. The DE2-115 FPGA board was used [23]. The use of an FPGA board was confirmed to be necessary after a 16-bit microcontroller was found to have insufficient processing power for an advanced vision system. This was determined from the performance of an initial machine vision system using a Freescale Smart Car.

The Freescale Smart Car competition [31] involved designing a line following system under varied lighting and on slopes. Firstly, the microcontroller was the bottleneck in the system. It could not cope with the full capacity of the Mitsubishi M64282FP CMOS image sensor used by the GameBoy™ back in 1998 [30]. This camera has a resolution of only 120×120 pixels at 30 frames per second. Using the microcontroller, and an optimised algorithm, only 20 frames per second were possibly whilst reading 25 lines of 61 pixels each. This bottleneck strongly suggested and validated the use of an FPGA for the iSAT machine vision system. Microcontrollers were too slow and had too little memory.

Resultantly, a digital data path spanning input from the camera and rotary encoders through real-time vision processing to differential drive motor control output was developed using FPGA architecture.

Auxiliary to the main data path, a VGA monitor display of the FPGA image captured was designed. Feedback via the User Datagram Protocol (UDP) over Ethernet LAN using the lightweight Internet Protocol (lwIP) stack [39] was also integrated. This enabled proof of the processing capability to be obtained graphically. A remote control was also interfaced to allow the user to control the vehicle directly. Accompanying software was written for the host-PC using Visual C#.NET to enable developers to observe the video feed and the FPGA's processing of the images over a local area network.

The data path includes image capturing and both pre-storage and post-storage processing of the image. The central component is the memory-based arbitrator. Each part of the data path is considered in the sub-sections that follow.

4.8.1. Camera Interface for Image Acquisition

The camera interface was built with the aid of the datasheet [37]. The OV7620 has a 16-bit data bus, of which 11 pins were connected to meet the RGB565 pixel storage format. The camera is controlled by an I2C bus. Green pixel information was available for each of the 320×240 pixels on the *Y*-bus, whilst red and blue information was subsampled to 160×240 and available alternately on the *UV*-bus as illustrated in *Figure 4*.

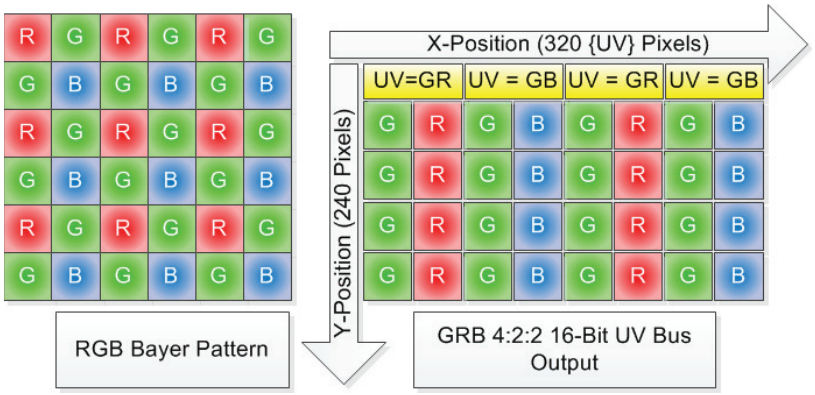

Fig. 4. Image sensors use a Bayer filter to extract the colour information from the scene being viewed. The selected output mode from the OV7620 camera was a 16-bit data stream of 4:2:2 GRB compressed pixel information as shown in the adapted illustration above [37]. Two neighbouring pixels share the same red and blue components but have unique green components.

4.8.2. Pre-Storage Image Processing

In the pre-storage path, colour conversion is performed using equations 1-3. These were modified from [40] to work on a 0-255 scale which is suitable for hardware implementation.

$$V_{0\to255} = \max(R,G,B) \tag{1}$$

$$S_{0\to255} = \begin{cases} 0 & if\ V = 0 \\ 255 - \dfrac{\min(R,G,B)}{\max(R,G,B)} & otherwise \end{cases} \tag{2}$$

$$H_{0\to255} = \begin{cases} \dfrac{3}{128}\dfrac{G-B}{\max(R,G,B)-\min(R,G,B)} & if\ R > G \geq B \\ \dfrac{3}{128}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)} + 256 & if\ R > B > G \\ \dfrac{3}{128}\dfrac{B-R}{\max(R,G,B)-\min(R,G,B)} + \dfrac{256}{3} & if\ \max() = G \\ \dfrac{3}{128}\dfrac{R-G}{\max(R,G,B)-\min(R,G,B)} + \dfrac{512}{3} & if\ \max() = B \end{cases} \tag{3}$$

At this stage, part of the image processing for the benchmark algorithm was included, namely the process of histogram binning [17]. Using the HSV values stored as HSV466, a subsample of the image, sized 40×30 pixels was selected from the centre of the image nearest the trolley. Binning was performed to group the data into 8 equispaced colour categories per component (H-S-V).

Following the recommendations [17] it was noted that hue values are only valid if there is significant saturation and saturation values are only valid below a maximum intensity value. As such, the intensity ceiling was set at 80% and the saturation floor at 5%. The hue data from a CMOS camera is typically noisy, so was given less priority. Several combinatory functions were examined such as (H|S|V) or (H&S|V) and finally (S|V) which provided the best results as illustrated by the MATLAB simulation in *Figure 5*. This was the algorithm targeted for implementation on the FPGA.

4.8.3. Memory-Based Arbitration

The main component of the data path architecture was the memory-based arbitration scheme. Leveraging upon the on-chip SRAM, the arbitrator was modelled around the concept of operating system scheduling.

The arbitrator was designed to have 28 time slots, each lasting 20ns. As part of a 4 stage pipeline, a preliminary slot of 10ns is allocated to requesting an address from an intellectual property (IP) block. At the following clock, the address is written to the RAM. If the operation is to write to memory, data from the IP is sent as well. At the third clock, data is read and forwarded to the peripheral in the case of a read operation. Finally, the fourth clock edge resets the peripheral for the next operation. A timeline is shown in *Figure 6* and the arbitration scheme developed is summarised in *Table 1*.

Fig. 5. MATLAB simulation of the colour histogram binning technique modified from [17]. Images (a) through (h) show the steps. The red line on image (i) shows the correctly identified limit of the floor with minor noise. This was achieved using a composite of saturation and value components. Refer to [17] for the full algorithm.

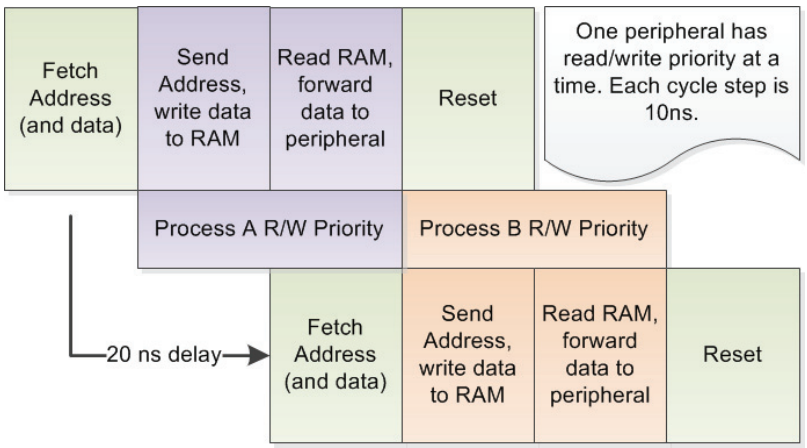

Fig. 6. Each process is scheduled as frequently as required across 28 available slots. Each sequence lasts 40ns with 10ns each allocated to fetching, writing, reading and resetting. The same process cannot be requested back-to-back.

The *Camera Pixel Acquisition Slot* accepts data from the pre-storage image processing. This slot buffers a live image acquisition stream into on-chip FPGA block RAM. Two blocks of RAM are activated alternately to act as a two-stage frame buffer. In this way, whilst a frame is being loaded into *RAM Block A*, *RAM Block B* will be relaying data to the post-storage processing block via *Slot 1 for Image Processing*.

Table 1. Arbitration specifications

Task	Deadline	Allocation
Camera Pixel Acquisition	148 ns	4/28
VGA Monitor Pixel Display	40 ns	7/28
Image Processing 1	148 ns	4/28
Image Processing 2 (free)	148 ns	4/28
Image Processing 3 (free)	148 ns	4/28
Image Processing 4 (free)	148 ns	4/28
Auxiliary (free)		1/28

Currently only the slots for camera, monitor and image process 1 are used. The remaining cycles are surplus capacity for expansion. Each cycle lasts 20ns and is part of a 4 stage 10ns pipeline.

4.8.4. Post-Storage Image Processing

The *Image Processing 1 Slot* in *Table 1* was used to fetch pixels from memory for post-storage processing. The post-storage processing uses the result from binning carried out during acquisition. Bins which are filled above a threshold (e.g. 50 pixels) are described as belonging to the background. Image Process 1, evaluates the entire image vertically, column-wise to establish the end of the floor. This data is then availed to the NIOS processor via an Avalon slave interface.

5. Results

The prototype design is summarised in *Table 2*. Results are provided to illustrate the successful design of the data path. As a proof-of-concept, results of FPGA-based image processing using this custom-built system are discussed.

5.1. Trolley Design and Performance

The trolley was designed to be rugged and a safety factor of 2 was maintained, enabling it to withstand static loads of 100kg. *Table 2* describes the tested specifications.

Table 2. Final design technical parameters

Parameter	Design Specification
Dimensions	L51xW51xH78 cm with 4cm ground clearance
Weight	20 kg including battery
Payload	50 kg not exceeding a height of 20cm
Body	Profiled aluminium sections with custom parts
Sensors	1 x OV7620 CMOS colour camera 2 x Hall Effect Rotary encoders for odometry
Image acquisition	320×240 pixels @ 60fps in GRB 4:2:2
Image processing	6 processes per pixel using Verilog HDL
Primary controller	Cyclone IV on DE2-115 FPGA Board
Interfaces	UDP over Ethernet LAN, Infrared Remote Control, VGA Monitor
Drive system	2 wheel differential drive
Travelling Speed	1m/s fully loaded, 2m/s at self-weight

5.2. Evaluating the Data Path Architecture

The camera input was observed to become clear after configuration using a custom-built hardware-based I2C controller to perform initialisation. The *VGA Monitor Slot* was interfaced to a purpose-built 640×480@60fps up-scaling VGA driver and connected to an LCD display. The successful confirmatory test output is shown in *Figure 7*. This test indicates the correct behaviour of the RAM-based arbitrator and pre-storage processing as no pixels are missing and the image is not distorted.

Fig. 7. The clear image on the monitor illustrates the successful capturing of the video stream from the camera and the high performance of the arbitrator designed in hardware. The image was rotated by 180° so that if the vehicle moves quickly, the same area would not be scanned repetitively.

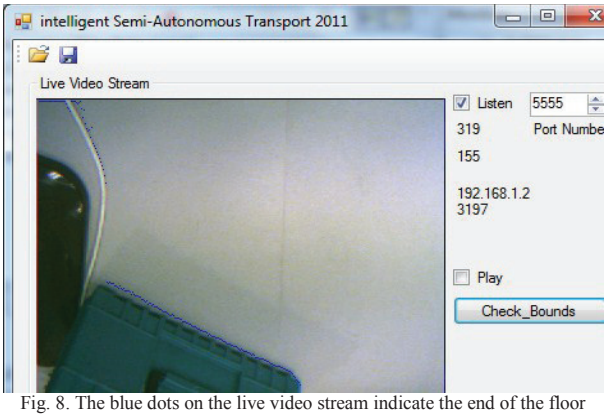

Fig. 8. The blue dots on the live video stream indicate the end of the floor from the top of the image. The dots were transmitted as an array from the hardware-based image processing block written in Verilog HDL and linked via a self-built Avalon interface to the NIOS processor and the custom code written in C using the lwIP stack to packetise the data. A data transmission rate of 15.8Mbit/s was recorded over Gigabit LAN.

5.3. Evaluating the FPGA-based Machine Vision System

An application was successfully developed on the NIOS processor using the lightweight Internet Protocol (lwIP) [39]. Similarly a custom-developed Visual C#.NET application was written on the host-PC. In this way an end-to-end UDP communication channel was established.

Using this channel it was possible to receive both the live image from the iSAT platform, as well as the outcome of the machine vision algorithm implemented in hardware that was derived from [17]. A screen-shot is shown in *Figure 8*. As can be seen by the blue dots overlaid on top of the live image feed, the vision algorithm is performing optimally, as the floor boundary is being clearly marked, whilst the open floor space remains dot-free.

6. Conclusion

Difficulties were encountered in implementing the data path, largely due to digital timing failures during synthesis. These were time-consuming to rectify. Similarly, as a low-level CMOS camera was interfaced, significant effort was required in order to fully adhere to the camera-port specifications. The same applied to the VGA standard for displaying output on the LCD monitor.

The primary objective of using hardware-software co-design to develop a real-time machine vision system for a mobile robot has been successful. The data path developed performs well with deadlines as short as 40ns being met; something unlikely to be achieved using microcontrollers.

The secondary objective of developing a realistic trolley prototype was also successful as the device is sturdy and powerful. As future work, hardware compression could be applied to the image which would enable a wireless link to be integrated instead of Ethernet LAN. A digital compass could be used to complement odometry. The data path avails excess capacity that could be used for more advanced vision algorithms such as visual Simultaneous Localisation and Mapping. A path planner could also be implemented.

References

[1] Davies ER. Machine Vision: Theory, Algorithms, Practicalities. 3rd ed. MO: Elsevier; 2005.
[2] Schalkoff RJ. Intelligent Systems: Prinicples, Paradigms and Pragmatics Sudbury: Jones & Bartlett; 2011.
[3] Guiness World Records. Busiest Airport for International Passengers. [Online]; 2008 [cited 2011 May 29]. Available from: http://www.guinnessworldrecords.com/search/Details/Busiest-airport-for-international-passengers/53512.htm.
[4] Wang KE, Wang W, Zhuang Y. A Map Approach for Vision-based Self-localization of Mobile robot. Acta Automatica Sinica. 2008 February; 34(2): p. 159-166.
[5] Sidla O, Brandle N, Beensova W, Rosner M, Lypetskyy Y. Embedded Vision Challenges. In *"Smart Cameras"* Belbachir AN, editor. New York: Springer; 2010. p. 99-117.
[6] Batlle J, Marti J, Riadao P, Amat J. A New FPGA/DSP-based Parallel Architecture for Real-Time Image Processing. Real-Time Imaging. 2002 February; 8(1): p. 345-356.
[7] Ubbens TW, Schuurman DC. Vision-Based Obstacle Detection using a Support Vector Machine. In IEEE Canadian Conference on Electrical and Computer Engineering. CCECE '09.; 2009; St. John's, NL. p. 459-462.
[8] Benmessaoud ML, Lamrani A, Nemra K, Souici AK. Single-Camera EKF-vSLAM. World Academy of Science, Engineering and Technology. 2008; 47.
[9] Karlsson N, Bernardo ED, Ostrowski J, Goncalves L, Pirjanian P, Munich ME. The vSLAM Algorithm for Robust Localiation and Mapping. In Proc. of Int. Conf. on Robotics and Automation; 2005; Barcelona.
[10] Surveyor Corporation. Surveyor SRV-1 Blackfin Camera. [Online]; 2010. Available from: http://www.surveyor.com/blackfin/.
[11] Carnegie Melon University. CMUcam. [Online]; 2011 [cited 2011 May 31]. Available from: http://cmucam.org/.
[12] Hamblen J, Hall T, Furman M. Rapid Prototyping of Digital Systems. SOPC Edition ed. New York: Springer; 2008.
[13] Intel Corporation. Intel PXA270 Processor, Electrical, Mechanical & Thermal Specification. [Online]; 2005. Available from: http://www.phytec.com/pdf/datasheets/PXA270_DS.pdf.
[14] Texas Instruments. OMAP Applications Processor. [Online]; 2010. Available from: http://focus.ti.com/docs/prod/folders/print/omap3530.html.
[15] Witech. Arm9Board. [Online]; 2010 [cited 2010 October 17. Available from: http://www.arm9board.net/sel/prddetail.aspx?id=348&pid=200.
[16] Blackham B. Development of a Hardware Platform for Real-Time Image Processing. Final Year Project. University of Western Australia; 2006.
[17] Ulrich I, Nourbakhsh I. Appearance-Based Obstacle Detection with Monocular Color Vision. In AAAI National Conference on Artificial Intelligence; 2000; Austin. p. 866-871.
[18] Kang JS, Jeon J, An Y, Park J. Mobile Vision Robot Using the SIFT Algorithm for Object Recognition. In IEEE International Conference on Ultra Modern Telecommunications; 2009; St. Petersburg. p. 1-7.
[19] Bouris D, Nikitakis A, Papaefstathiou I. Fast and efficient FPGA-based feature detection employing the SURF algorithm. In Field-Prgrammable Custom Computing Machines; 2010; Charlotte: IEEE. p. 3-10.
[20] Konolige K, Bowman J, Chen JD, Mihelich P, Calonder M, Lepetit V, et al. View-Based Maps. Robotics Research. 2010 July; 29(8): p. 941-957.
[21] Greco J. Parallel Image Processing and Computer Vision Architecture. Undergraduate Thesis. University of Florida; 2005.
[22] Asano S, Maruyama T, Yamaguchi Y. Performance comparison of FPGA, GPU and CPU in Image Processing. In International Conference on Field Programmable Logic and Applications; 2009; Prague: IEEE. p. 126-131.
[23] Terasic. Altera DE2-115 Development and Education Board. [Online]; 2011 [cited 2011 May 30]. Available from: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=502.
[24] Advantech. PCM-3362 Support Portal. [Online].; 2011 [cited 2011 May 31. Available from: http://support.advantech.com.tw/support/DownloadSearchByProduct.aspx?keyword=PCM-3362.
[25] nVidia Corporation. nVidia CUDA Platform. [Online].; 2010. Available from: http://www.nvidia.com/object/what_is_cuda_new.html.
[26] Technologic Systems. Technologic Systems PC/104 Single Board Computers and Peripherals. [Online].; 2010. Available from: http://www.embeddedarm.com.
[27] Wiseman S. Balloon Homepage. [Online]; 2010. Available from: www.balloonboard.org/.
[28] Prevas. [Online]; 2007 [cited 2010 October 17]. Available from: http://www.prevas.dk/download/18.337d8c3a10fe22dc03780009126/s1_brochure_dk.pdf.
[29] Berns K, von Puttkamer E. Autonomous Land Vehicles: Steps towards service robots. 1st ed. Netherlands: Springer; 2009.
[30] Herrington D. Atmel. [Online]; 2005 [cited 2010 October 18]. Available from: http://www.atmel.com/dyn/resources/prod_documents/issue4_pg39_43_robotics.pdf.
[31] Freescale Semiconductor. The Freescale Cup. [Online]; 2011 [cited 2011 June 8]. Available from: http://www.freescale.com/webapp/sps/site/homepage.jsp?code=SMARTCARHOME.
[32] Electronics 123. Electronics 123. [Online]; 2010 [cited 2010 October 17]. Available from: http://www.electronics123.net/amazon/datasheet/c3188a.pdf.
[33] Shayang Ye Industrial. Shayang Ye. [Online]; 2010 [cited 2010 Ocober 17]. Available from: http://www.shayye.com.tw/english/products.html.
[34] Cytron Technologies. Enhanced 30 Amp DC Motor Driver. [Online]; 2011 [cited 2011 May 31]. Available from: http://www.cytron.com.my/viewProduct.php?pid=IQImJRsKHSA1NQ8fAgIXLGEs7juPWoKF5Jyb/MvJ/9I=.
[35] Craig K. Multidisciplinary Mechatronics Innovations, Marquette University. [Online]; 2011 [cited 2011 June 5]. Available from: http://www.multimechatronics.com/.
[36] Yan XT, Zante R. A Mechatronic Design Process and its Application. In Bradley D, Russell DW. Mechatronics in Action: Case Studies in Mechatronics - Applications and Education. Londom: Springer-Verlag; 2010. p. 55-70.
[37] Omnivision. OV7620 Single-Chip CMOS VGA Color Digital Camera. [Online]; 2000 [cited 2011 May 31]. Available from: http://www.electronics123.net/amazon/datasheet/OV7620.pdf.
[38] Braunl T. Embedded Robotics: mobile robot design and applications. 3rd ed. Berlin: Springer; 2008.
[39] Dunkels A. LWIP: A Lightweight TCP/IP Stack. [Online]; 2011 [cited 2011 May 31]. Available from: http://savannah.nongnu.org/projects/lwip/.
[40] Melin P, Kacprzyk J. Soft Computing for Recognition Based on Biometrics Pedrycz W, editor. Chennai: Springer; 2010.