3rd International Conference on Sensing Technology, Nov. 30 – Dec. 3, 2008, Tainan, Taiwan

Heterodyne Range Imaging in Real-time

A. P. P. Jongenelen[1], D. A. Carnegie[1], A. A. Dorrington[2] and A. D. Payne[2]
[1]School of Chemical and Physical Sciences, Victoria University of Wellington, Wellington, New Zealand.
[2]Department of Engineering, University of Waikato, Hamilton, New Zealand.
adrian.jongenelen@vuw.ac.nz

Abstract

A versatile full-field range imaging system has previously been constructed. This system is configurable in software to produce either high precision or fast acquisition range images. Indicatively a 10 second exposure has been shown to produce a range image of sub-millimeter precision, whilst video frame rate (30 fps) acquisition provides for centimetre precision.

Currently the acquisition time of the system is to a large degree constrained by the off-line processing of the frames by an external computer. This paper presents an alternative to the off-line PC image processing utilising an Altera Stratix II FPGA. Processing rates up to 30 frames per second have been achieved with the added advantage that many of the previous system's existing digital electronics can also be accommodated, providing for an even more compact and flexible system.

Keywords: Range image, ranging, real-time, heterodyne, FPGA

1 Introduction

A range imaging camera produces a two dimensional array of pixels where in addition to representing the colour or intensity of an object in the scene, the distance from the camera to the object is also encoded [1, 2]. This allows for a construction of a three dimensional model of the scene, useful in applications such as machine vision, animation and process control.

A solid-state range imaging system constructed from off-the-shelf components has been set up in a heterodyning configuration [3, 4]. This system has the ability to be easily configured through software to take long exposure, high resolution depth images with sub-millimetre precision or video frame rate depth images at the expense of precision.

At present the system is limited in that raw frames of intensity data are transferred from the camera to a PC where depth images are calculated in an offline process. In a real world application of this technology the end user would be presented only with depth and intensity information, and the intermediate frames captured by the camera are not useful. Therefore it is ideal if the system can pre-process the data and extract the depth information in real-time, relieving the end user of this task so that more effort can be put into higher level processing of the data.

In this paper we present an extension to the existing system where processing is performed in real-time using an Altera Stratix II FPGA at output frame rates of up to 30 frames per second. Section 2 describes the novel heterodyning technique used by our ranging system and Section 3 summarises how this has been achieved using off-the-shelf components. A description of the extensions to the system to implement real-time range processing are presented in Sections 4 and 5 with experimental results and discussion in Section 6.

2 Heterodyne Ranging Technique

Our range imaging system uses a heterodyning time-of-flight method for measuring the range of objects in the field of view. This is achieved by projecting intensity modulated light onto the scene at a modulation frequency f_M, and shuttering the light received by the camera at a frequency offset by a known amount, $f_M + f_B$. These signals are effectively mixed and integrated by the camera to produce a beat signal of frequency f_B.

Additionally, the light received at the camera has a phase offset introduced due to the time taken for the light to travel twice the distance from the camera to the object. Consequently the problem of calculating the distance to the object is simplified to calculating the phase of a signal at the much lower beat frequency. Since the beat frequency is fixed and known, this can be achieved using a single bin Discrete Fourier Transform [5]. If an integer number of beat cycles are sampled, the phase offset, θ, can be calculated as the inner product

$$\theta = \arctan\left(\frac{\sum I_i \cos(\frac{2\pi i f_B}{f_S})}{\sum I_i \sin(\frac{2\pi i f_B}{f_S})}\right) \tag{1}$$

where I_i are the samples of pixel intensity and f_S is the sampling rate of the camera. The distance from the object to the camera, d, is then calculated as

$$d = \frac{c\theta}{4\pi f_M} \tag{2}$$

where c is the speed of light.

978-1-4244-2177-0/08/$25.00 © 2008 IEEE

3 The Existing Ranging Hardware

The existing hardware setup is shown in Figure 1.

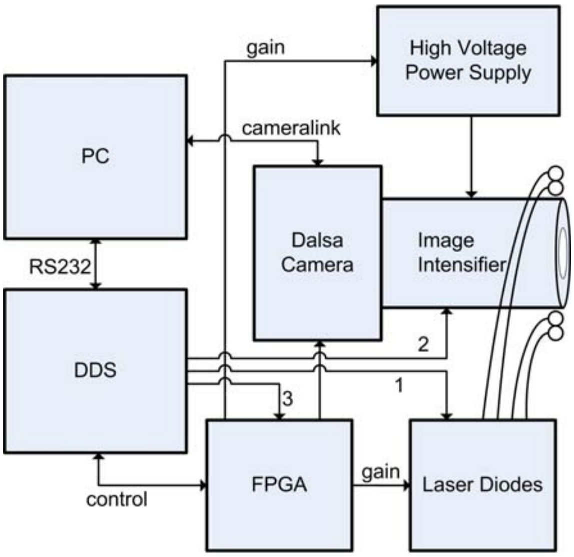

Figure 1: Existing Ranger System hardware configuration

The scene is illuminated by four 80 mW laser diodes intensity modulated at the base frequency f_M. The camera used is a Dalsa Pantera 1M60 digital video camera, capable of frame rates up to 220 Hz with a resolution of 128 by 128 pixels. To achieve the high frequency shuttering, the camera is coupled to a Photek MCP125 Image Intensifier with the modulation signal applied to the photocathode of the image intensifier.

For the heterodyning setup described in Section 2 it is imperative that the modulation signals driving the shuttering of the image intensifier and the light source have a stable low frequency difference. Furthermore, in order to calculate absolute range measurements it is necessary that the phase difference is known at any point in time during a frame capturing sequence.

To meet these requirements a circuit board has been constructed containing three Analog Devices AD9952 DDS ICs [6]. An advantage of these ICs is their ability to be synchronised to a single clock, ensuring minimal phase error between devices. Each IC has an internal 400 MHz system clock generated by multiplying a 20 MHz reference temperature-compensated oscillator.

To generate a desired output frequency the DDS is programmed with a 32-bit Frequency Tuning Word, FTW, using the SPI interface of a 89LS8252 microcontroller connected through a RS232 serial link to a PC. This is calculated as

$$FTW = \frac{2^{32} f}{f_{SYS}} \tag{3}$$

where f is the desired frequency and f_{SYS} is the 400 MHz DDS system clock frequency.

The three sinusoidal frequencies generated by the DDS are used for: 1) the transmitted light, f_M, 2) the shutter of the camera sensor, $f_M + f_B$ and 3) the frame rate of the camera, f_S. Each channel has a resolution of 0.093 Hz as calculated by Equation 3 with an FTW of 1, and a maximum frequency of 160 MHz limited by a low-pass filter on the output. An additional CMOS output is provided by the DDS to signal when all three output signals are in phase. This is necessary to calculate absolute range data as opposed to only relative range.

The frequency provided by the DDS board for the camera trigger input is multiplied by a constant, giving mf_S. This reference signal is then passed into a digital counter to divide the frequency back down to the sampling rate, reducing jitter arising from converting the DDS sine wave output to the digital CMOS input required for the camera.

A Xilinx Spartan 2 FPGA is used to divide the frequency of the 3rd channel back down to f_S. In conjunction with the microcontroller on the DDS board and a pair of DAC ICs, the FPGA also controls the analogue gain of the laser diodes and image intensifier. In the case of the image intensifier, this is used to ensure that the pixels of the camera do not saturate. For the laser diodes it allows them to be initially turned on over a period of several minutes, protecting them from damage caused by overheating.

The PC provides control of the modulation frequency and beat frequency through the RS232 connection with the DDS board. Frames captured by the camera are transferred to the PC via the industry standard Cameralink [7] interface, and subsequently processed offline.

4 Hardware Processing Algorithm

To translate Equation 1 into a hardware implementation, each basic operation is given a dedicated block of FPGA hardware. Equation 1 can be expanded out as shown below, labelling all intermediate results.

$$\begin{aligned} new\_r &= raw \times cos(i) \\ new\_i &= raw \times sin(i) \\ real &= new\_r + real \\ imag &= new\_i + imag \\ div &= imag\ /\ real \\ atan &= atan(div) \end{aligned} \tag{4}$$

Figure 2 shows how this is realised in the hardware. The purpose of the block RAM is to store the previous result from the multiplication, for all pixels of the frame. This RAM is dual-port, allowing for new values to be written in and old values to be read out concurrently. The values of $cos(i)$ and $sin(i)$ are taken from a series of registers which can be modified by the user in order to change the number of sample points per beat cycle. The index i is a counter incremented after each new frame is processed. The look-up-table (LUT) for the arc tangent function is configured as a block of read-only RAM.

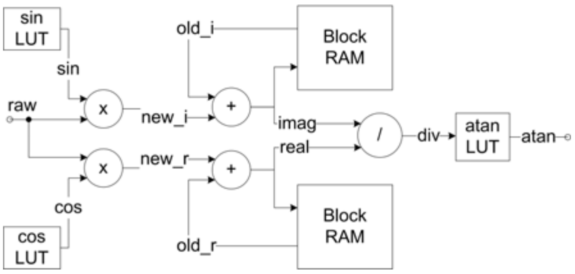

Figure 2: Conversion of Inner Product Equation into functional hardware blocks

The user also has control over how many frames, n, are expected per beat cycle. It has been shown that an n value of 5 provides a good compromise between precision and short acquisition time [8]. Not shown is the reset circuitry to reset the i counter and the old_r and old_i values once n frames have been processed. For the previous values this is achieved by testing if the counter is at zero, and then ignoring the previous value and using zero instead. This results in an output frame rate, f_O, of

$$f_O = \frac{f_S}{n} \tag{5}$$

A sliding window approach, whereby the last n number of frames are used to calculate the output, can increase the output frame rate to be equal to f_S. However, this is much less flexible in terms of changing the frames per beat since much more block RAM must be preallocated to store n number of frames. Also, although it increases the output frame rate, it does not increase the output time response. That is, it will still take n frames for a step change in the scene to be accurately realised in the output.

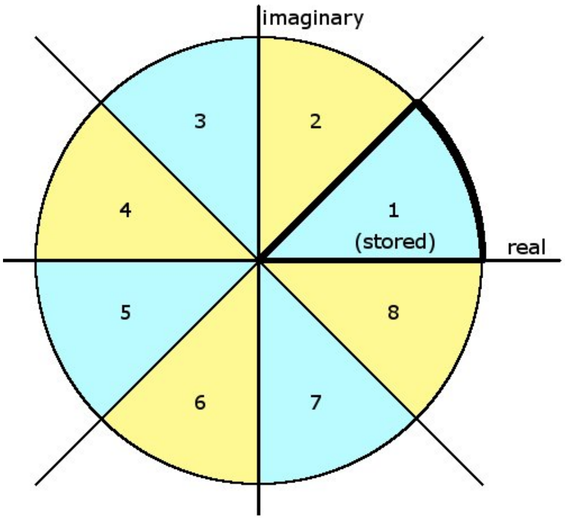

Figure 3: Regions for calculating arc tangent function

The symmetry of the arc tangent function can be taken advantage of in order to keep the FPGA resource usage for the function to a minimum. A coarse estimate of the arc tangent output is first obtained based on the sign and relative magnitudes of the real and imaginary inputs, and it is then established in which of the eight sectors of Figure 3 the result will lie. A division on the absolute values of the inputs is performed such that the denominator is always larger than the numerator, and the intermediate result labelled x. Only the arc tangent values from 0 to $\frac{\pi}{4}$ are stored in the LUT, and the final result is calculated by adding or subtracting this value to or from a constant as per Table 1.

Table 1: Calculation of arc tangent using 0 to $\frac{\pi}{4}$ look-up-table

Sector	*Division*	*Addition/Subtraction*
1	$x = \frac{\lvert imag \rvert}{\lvert real \rvert}$	$\theta = atan(x)$
2	$x = \frac{\lvert real \rvert}{\lvert imag \rvert}$	$\theta = \frac{\pi}{2} - atan(x)$
3	$x = \frac{\lvert real \rvert}{\lvert imag \rvert}$	$\theta = \frac{\pi}{2} + atan(x)$
4	$x = \frac{\lvert imag \rvert}{\lvert real \rvert}$	$\theta = \pi - atan(x)$
5	$x = \frac{\lvert imag \rvert}{\lvert real \rvert}$	$\theta = atan(x) - \pi$
6	$x = \frac{\lvert real \rvert}{\lvert imag \rvert}$	$\theta = -atan(x) - \frac{\pi}{2}$
7	$x = \frac{\lvert real \rvert}{\lvert imag \rvert}$	$\theta = atan(x) - \frac{\pi}{2}$
8	$x = \frac{\lvert imag \rvert}{\lvert real \rvert}$	$\theta = -atan(x)$

The advantage of this method is that the result from the division is always between 0 and 1 which can scale easily as an index into RAM. Also, the LUT has only one eighth of the memory requirements. With a 1024 × 16-bit LUT the arc tangent function can be approximated to within 0.5 milliradians.

A fixed-point number representation is used throughout the process, starting at 12 bits wide for the raw input from the camera, and expanding to 22 bits wide for the input to the division operator within the arc tangent function block. The process is tested by generating a series of test frames where each pixel has a different phase and magnitude. The overall error introduced compared with a 64-bit floating point process (as previously used in the Matlab based processing) is approximately ±2 milliradians.

The raw data is streamed directly from the Dalsa camera with a pixel clock of 40 MHz per output tap. The camera has two taps, one for each half of the imaging area, which are processed independently by instantiating the ranging process two times within the FPGA, one for each tap. The process is fully pipelined with a latency of 24 clock cycles from new pixel data being received to the result appearing from the arc tangent function. Targetting a speed grade 3 Altera Stratix II EP2S60 IC a theoretical maximum frequency of 144 MHz is predicted by the synthesis software.

The final result is stored in two areas of dual-port block RAM, one to be accessed by the Altera Nios CPU core which handles sending the data back to a PC over the ethernet, and the other is accessed by a VGA monitor driver for real-time display.

5 FPGA System Top Level

The target IC for this design is an Altera EP2S60 FPGA mounted on an Altera Nios II Stratix II (RoHS) Development Kit [9]. Key advantages that this board offers are:

- Stratix II EP2S60 FPGA
  - 144 18-bit × 18-bit hardware multipliers,
  - 12 Phase-locked-loops (PLLs),
  - 2,544,192 bits block RAM,
  - 60,440 equivalent logic elements,
- Nios II soft processor core with 32 MB DDR SDRAM,
- 10/100/1000 ethernet port,
- JTAG programming and debugging port,
- RS232 serial port,
- Two general purpose 41-pin expansion headers.

Figure 4 gives an overview of the system including connections to the Dalsa camera, the PC and a standard VGA monitor for real-time display. The PC is no longer involved in the processing of the depth images and is included purely as a means for long term storage. Additional boards have been produced and plugged in to the expansion headers to drive the VGA monitor and interface to the camera using the Cameralink interface.

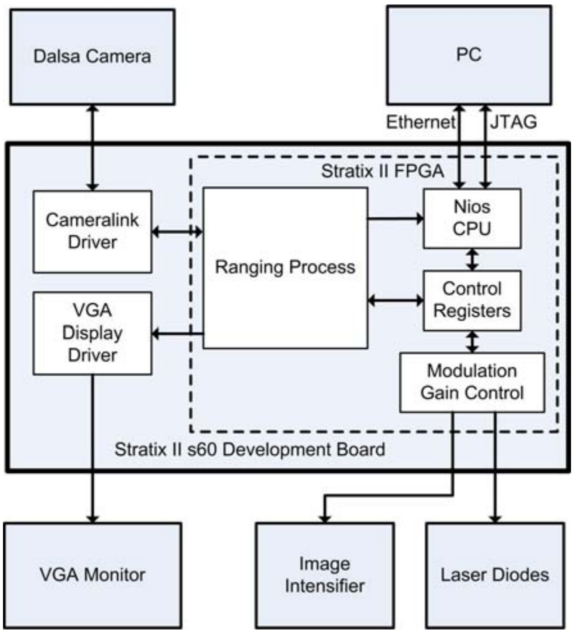

Figure 4: Altera Stratix II Development Kit and interconnections

Frames are captured by the Dalsa camera and transferred directly to the FPGA through the Cameralink interface. These are processed in real-time with resultant frames stored on the FPGA. These frames are then output to the VGA monitor and retrieved by the Nios CPU to be transferred out to the PC via a TCP/IP connection on the ethernet port. Control registers for the ranging process are configured through the Nios CPU via either the JTAG debugging interface or the TCP/IP connection. Also shown in Figure 4 is the logical block for controlling the gain of the modulation signals to the image intensifier and laser diodes.

Software tools used in the design are Altera's Quartus II 7.1 for VHDL synthesis, and Nios II 7.1 IDE for programming of the Nios II CPU and JTAG communication. Aldec's Riviera Pro is used for VHDL simulation, and Microsoft Visual Studio is used to program the TCP/IP client program for the PC.

6 Results and Discussion

Overall, removal of the PC from the processing loop has been successful. The system is still flexible in its ability to select between high precision or fast acquisition time operating modes, and the end user such as a desktop PC or mobile robot's CPU is relieved of the task of calculating the depth data from the scene. Output frame rates of up to 30 frames per second have been achieved, with f_S = 150 Hz, n = 5 and camera resolution = 128 × 128 pixels.

Table 2 shows a summary of FPGA resources used by various elements of the design. The logic requirements of the design are not large, using only 9% of available Adaptive Logic Modules (ALMs), however the large block RAM requirement limits the pixel resolution of the processed images. For 128 × 128 pixels the 22-bit wide accumulated real and imaginary values account for 720,896 bits. The remainder of block RAM used in the ranger process represent the arc tangent LUT and the output buffers for the Nios CPU and VGA module.

Table 2: Summary of Stratix II EP2S60 FPGA Resource Usage

Resource	*ALMs*	*DSP9×9*	*BlockRAM (bits)*
Ranger Process	1529	8	1,163,476
Nios CPU	2285	8	48,128
Other	417	0	181,248
Total	4231	16	1,392,852
Total as % of available	9%	6%	55%

To experimentally verify the plausibility of the realtime ranging system a number of captures of dynamic scenes have been taken. The example described here involves a number of objects moving about in the scene: a pendulum swinging in a circular arc across the camera's field of view, a teddy bear figurine rotating on a turntable, and a roll of paper towels rolling down a ramp towards the camera. Modulation frequency for this capture is 40 MHz.

Figure 5 shows the scene at the start of the capture with depth represented by pixel intensity. Five test points are identified in the image: 1) a region of cardboard background above the bear which does not change, 2) a region of the far wall of the room, and 3) to 5) are points used to track the movement of the paper towels rolling down the ramp. The bright spot to the right of the bear shows the pendulum, although in this capture it moves too quickly for its depth to be analysed.

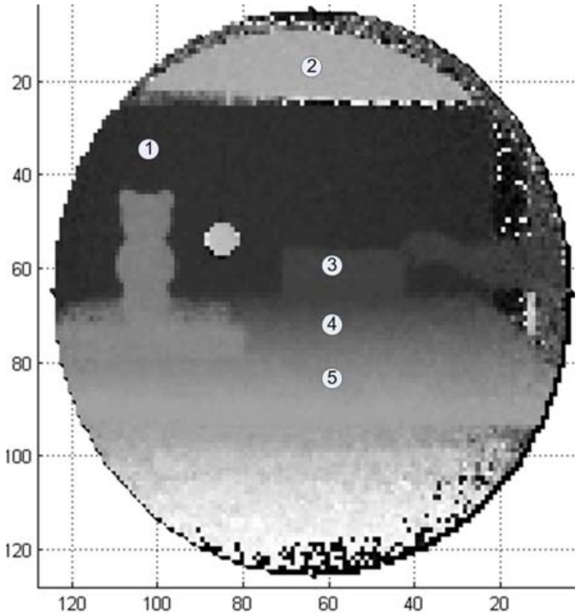

Figure 5: Frame 1 of test capture with identified test points

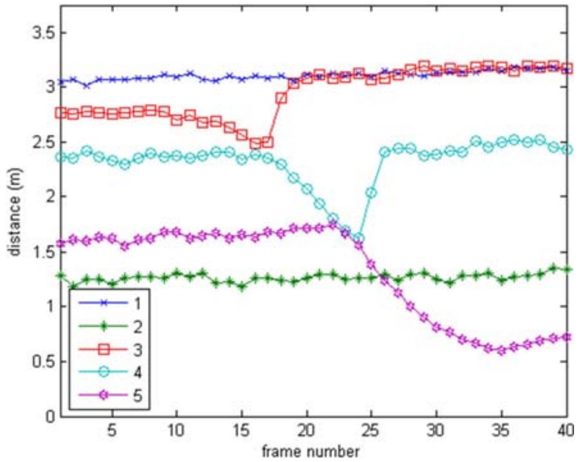

Figure 6: Range of test points throughout 40 frame capture

Figure 6 shows the calculated depth of the test locations for each frame of the capture. The plot shows the paper towels rolling towards the camera, and even shows them reaching the front and bouncing back. This can be seen from test points 3 to 5, where an interconnecting curve has a decreasing slope, suggesting the object is moving towards the camera. After the paper towels roll below test point 3 it can also be seen that this point then returns the same depth as test point 1, i.e., the background. Test points 1 and 2 show a fairly constant depth with point 1 having a mean of 3.11 m and standard deviation of 0.040 and point 2 having a mean of 1.27 m and standard deviation of 0.036. Test point 2 represents the far wall, but it returns a depth similar to where the paper towels pass through. This is due to the fact the back wall is beyond the unambiguous range of $c/2f_M$ = 3.75 m. In fact, without taking further measurements at a different modulation frequency none of the points can be absolutely reckoned beyond this distance.

The major bottleneck in the system is the method of handling the ethernet packets by the Nios CPU, generating a throughput of only 8 output frames per second, or 2.1 MBit/s back to the PC. However, this is handled well, in that while the output frame rate is lower, the frames are still processed at f_S/n, with regularly spaced output frames being dropped. For frames displayed on the VGA monitor, the full f_S/n frame rate is achieved up to 60 Hz.

Immediate future work on the system will involve the following:

- Improvement of the Nios ethernet packet handling to increase the output frame rate,
- Incorporation of low latency external SRAM to replace on-chip block RAM and provide for higher resolution image processing,
- Further characterisation of the precision of the real-time ranging process,
- Implementation of directly generating the modulation signals from the FPGA as described in [10] and removal of the DDS board,
- Performing automatic disambiguation by varying the modulation frequency on-the-fly.

7 Acknowledgements

The authors would like to thank Michael Cree for assistance with the processing algorithm and MATLAB prototyping.

Also, this research would not have been possible without funding through the William Georgetti Scholarship and a Tertiary Education Commission Top Achievers Doctoral Scholarship.

8 References

[1] F. Blais, “Review of 20 years of range sensor development”, *Journal of Electronic Imaging*, 13, pp 231–243 (2004).

[2] S. Christie, S. L. Hill, B. Bury, J. O. Gray and K. M. Booth, “Design and development of a multi-detecting two-dimensional ranging sensor”, *Measurement Science and Technology*, 6, pp 1301–1308 (1995).

[3] D. A. Carnegie, M. J. Cree and A. A. Dorrington, “A high-resolution full-field range imaging system”, *Review of Scientific Instruments*, 76, 083702 (2005).

[4] A. A. Dorrington, M. J. Cree, A. D. Payne, R. M. Conroy and D. A. Carnegie, “Achieving

sub-millimetre precision with a solid-state full-field heterodyning range imaging camera", *Measurement Science and Technology*, 18, pp 2809–2816 (2007).

[5] P. O'Shea, "Phase measurement", in J. G. Webster (ed.), *Electrical Measurement, Signal Processing and Displays*, CRC Press, Boca Raton, Fl, pp 41-48 (2003).

[6] A. Payne, D. Carnegie, A. Dorrington and M. Cree, "A Synchronised Direct Digital Synthesiser", in *International Conference on Sensing Technology*, pp 174–179, Palmerston North (2005).

[7] DALSA, *Pantera TF 1M60 and 1M30 User's Manual and Reference* (2004), www.dalsa.com.

[8] A. A. Dorrington, M. J. Cree, D. A. Carnegie, A. D. Payne, R. M. Conroy, J. P. Godbaz and A. P. P. Jongenelen, "Video-rate or high-precision: a flexible range imaging camera", *Image Processing: Machine Vision Applications. Proceedings of the SPIE*, 6813, 681307 (2008).

[9] Altera Corporation, *Nios Development Board Stratix II Edition Reference Manual*, v1.3 edition (2007), http://www.altera.com/products/devkits/altera/kit-niosii-2S60.html.

[10] A. D. Payne, *Development of a Full-Field Time-of-Flight Range Imaging System*, Ph.D. thesis, The University of Waikato (2008).