Prototype of High Speed Pipe Inspection Robot

Matthieu Jones, Donald Bailey, Liqiong Tang
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
mattyj_27@yahoo.com.au

***Abstract*-This paper presents the development of a high speed pipe inspection robot using structured light to measure erosion inside a pipe. Several factors influencing the speed are considered and an implementation based on a field programmable gate array (FPGA) is presented. This is a low power approach that would allow the processing of the high resolution images to be done in real time on the robot, with high processing speeds and rapid data reduction ideal for wireless transmission.**

I. INTRODUCTION

There are many constraints on a robot that is to be used for inspecting the inside of a pipe and many of these affect the speed in which the robotic system can perform its inspection. Some of these constraints include the imaging system used, the process used for data transmission, and the physical size, shape and design of the robotic platform.

There are several different methods that can be used for measuring the erosion in a pipe which directly affect the speed that a robot could inspect an entire pipe. It is common to use closed circuit television to inspect a pipe and this provides good qualitative detail on the actual state of the pipe. However, this requires a large amount of offline human analysis and generates a large amount of data which can be difficult to both transmit and store especially for long pipes. It is also difficult to directly access quantitative data on the erosion from such video data.

There are several approaches to actually measure the features of interest, such as erosion, to obtain quantitative data. These may include using an infrared thermography system, a laser transducer, ultrasonic sensors and pipe penetrating radar [1-4]. Mechanically measuring the depth at each point, for example using probes, was ruled out as being impractical due to the relatively slow speed. Video processing has the advantage of being non-contact, potentially enabling higher inspection speeds.

In this project, the feature of interest is the level of erosion in a straight, empty, circular concrete pipe. The required quantitative data may be obtained from video using structured lighting to measure a two-dimensional profile or cross-section of the pipe. A full three dimensional map of the pipe can be derived by exploiting the motion of the robotic platform within the pipe.

One of the main limitations of video based pipe inspection is the large volume of raw data which needs to be processed. Any non-trivial processing requires considerable computing power, which would significantly increase the volume and power requirements on the robot if a conventional computing platform was used. The transmission bandwidth of the raw video to a host processor places an upper limit on the resolution or inspection speed. The large data bandwidth is usually carried by running an umbilical cable between the robot platform and host. With such a cable, deployment can also limit inspection speeds.

The data volume may be reduced by pre-processing the video data using an embedded processing unit. With lower data volume, the need for an umbilical connection is reduced. Two alternatives are to either store the data locally or transmit it by some other means such as by using a wireless module. Different forms of communication have different transmission speeds and bandwidth which may pose limitations on the inspection system's resolution or processing speed.

Regardless of the communication system used, it is desirable to use some form of lossless compression to reduce the data volume. Any such compression would reduce the data bandwidth, which will have a positive effect on the speed at which the robot can inspect the pipe. However, any compression increases the computational burden on the mobile platform.

To achieve small size and low power, while achieving sufficient computational capacity, the embedded processing unit was based on a field programmable gate array (FPGA). This enables the video to be processed directly as it is streamed from the camera.

It is important to reduce the weight of the robot as much as possible as this will enable a smaller drive system to maintain the same speed. This is necessary as the robot must operate inside the pipe so is constrained by the physical features of the pipe. This will also impose other constraints such as the design and layout of the imaging system.

The development of the entire inspection robot is beyond the focus of this paper; rather, this paper focuses on the development of an imaging system to allow for a high speed inspection of a straight, empty, and circular, concrete pipe.

II. SYSTEM CONFIGURATION

The imaging system uses a structured light to obtain 3D information on the pipe surface. A single laser light stripe is projected onto the pipe wall and is imaged obliquely using a camera. Triangulation between the laser and the camera allows the three dimensional position of each point on the laser stripe to be calculated. The intersection of a plane (the

© 2012 ISBN: 978-0-473-20485-3

light stripe) with the circular pipe cross-section produces an elliptical curve in the image. Erosion will cause the position of the laser stripe to be offset, enabling the erosion to be measured.

The embedded processing unit is responsible for extracting the position of the laser stripe from the captured video stream, effectively reducing each 2D image to a 1D profile. This represents a significant reduction in the volume of data which must be transmitted.

There are two main configurations for setting up the camera and laser for achieving this; the first is to have the camera pointing vertically down with the laser on an angle, and the second configuration is the reverse of this, with the camera on an angle and the laser pointing vertically downwards. The second configuration was chosen for the reasons outlined in [5]. There the two configurations are compared and it is suggested that, while both have their comparative advantages and disadvantages, the second configuration is more appropriate for use in this application. This is because it enables a greater angle of view by having the camera mounted further away. It also makes it easier to obtain regularly spaced samples as the laser intersects the pipe at the same distance along the pipe regardless of any erosion present.

For this application the camera was mounted with the line of sight 40° from vertical, and the laser was mounted on a bracket cantilevered out in front of the camera and the rest of the robot as in Fig. 1. This creates the problem that if the robot is to be used in pipes with different diameters, a mechanism must be devised where both the distance between the camera and laser can increase as well as the vertical heights of each can be adjusted. Alternatively, different brackets could be used for different diameter pipes.

The first prototype imaging system was implemented using a Terasic DE0 development board [6] coupled with a 5 megapixel Terasic D5M camera module [7]. The DE0 board was chosen because of its compact size, and it has an integrated VGA output connection. The VGA output is essential during the initial development for debugging and tuning the image processing algorithms. Although the DE0 is an entry level board, it has more than sufficient resources for this application. The video display is not necessary after the initial development, so it is intended to use the smaller DE0 Nano board in the final prototype.

Figure 1: Setup showing FPGA camera on right with laser cantilevered on the left

Handel-C (version 5.5) was used to implement this algorithm with the resulting net-list mapped to the FPGA using Altera's Quartus II tools (11.1 Web Edition).

III. Image Processing

A. Algorithm Development

Image processing is used to extract the level of erosion across a cross-section of pipe. This is done by determining where the laser line is in the image. The reference profile corresponding to no erosion can then be subtracted from this and the result used to calculate the actual level of erosion in millimeters instead of pixels. This is non-trivial as it depends on the position in the image both vertically and horizontally.

In general, the pixel in each column with the greatest red component will correspond to the center of the laser stripe in the image. However, it is possible in situations where there is low reflectivity that the background may have a greater intensity. To avoid this, the green component of the image can be subtracted from the red. This will significantly increase the contrast by effectively removing the background of the image and having little effect on the laser stripe.

The laser stripe has a certain thickness in pixels and it is necessary to extract the center of the stripe. This can be done by filtering the image with a 12×1 averaging filter which will smooth the laser line out and produce a better defined peak corresponding to the center of the line. This allows a single pixel wide line to be extracted more reliably.

The line is then extracted by determining the row number that corresponds to the maximum pixel value in each column. This gives an accuracy to the nearest pixel, however this could be found to subpixel accuracy if this was required.

It is possible for occlusions to occur, due to the arrangement of the camera and laser, whereby the camera cannot actually see the laser in the image. Potential occlusions are identified by comparing the intensity of the maximum value in each column to a preset threshold. If it is below this threshold, and the position of the maximum value in the image is too far from the expected location of the laser, it is flagged as a potential occlusion.

The reference level can be subtracted from the resulting profile and the erosion profile can be determined by using the equation

$$d = \frac{\Delta\hat{y}}{k_1\Delta\hat{y} + k_2} \tag{1}$$

as determined in [8]. Where d is the level of erosion in millimeters, $\Delta\hat{y}$ is the level of erosion in pixels obtained from subtracting the reference, and k_1 and k_2 are constant for each column and can be predetermined.

B. High Speed System

The algorithm adopted has been designed for high speed implementation. This is achieved by keeping the image processing relatively simple and allowing it to be applied in real-time. Processing the image in real time means that it is not necessary to write the whole image to memory and then read it back while processing. Instead, the processing can be performed on the data as it is streamed from the camera so the processing on a frame can be completed almost as soon as the frame itself is captured.

To aid the image processing, the algorithm has been designed so that it can, if the hardware supports it, be implemented in parallel. That is, the processing, such as the averaging filter, can be applied to each column in unison instead of consecutively. This significantly reduces the latency, although at the expense of a small amount of memory to maintain the processing state for each column.

This algorithm also substantially reduces the volume of data. This reduces the data that needs to be processed, stored, and transmitted which significantly increases the rate of inspection.

C. Challenges on high speed implementation

While the processing is relatively simple, there is far too much data to process in real time using a low power embedded microcontroller. This can be overcome by processing the images in hardware, using an FPGA.

The main advantage of using an FPGA is the ability to implement the algorithm in parallel [9-10]. This is generally impossible using a microcontroller as they are inherently serial based processors.

IV. FPGA Based Algorithm

A. Prototyping (DE0)

The VGA output of the DE0 was used to display the output from the camera and the image processing at various stages and this requires a clock of 25 MHz. However, the camera has a maximum pixel clock of 96 MHz and it is desirable to run this as high as possible in order to increase the frame rate. Fortunately the DE0 has the ability to run different clock domains at the same time and has an onboard 50 MHz oscillator. A 25 MHz clock domain was set up to run the VGA by dividing the system clock by two, and a 86 MHz domain was set up for the camera and image processing using a PLL (phase locked loop). A clock of 86 MHz was used instead of 96 MHz (the maximum clock speed for the camera) because of the limitations imposed using a ribbon cable between the camera and the FPGA. Using different clock domains does introduce some difficulties. For example memory can only be read from or written to from one clock domain. Transferring data from one domain to another can be achieved either through the use of channels or by using dual-port memory, with one port in each clock domain. Channels are a useful structure for synchronizing different clock domains.

The images can be processed directly as they are streamed from the camera instead of capturing an entire image first. This permits several advantages as discussed previously. The logic for processing the images in this manner is outlined in Fig. 2.

The D5M camera has a 2592×1944 CMOS sensor with 12 bits per pixel output. The entire image does not need to be read out. Instead, a rectangular window can be taken of the relevant area in order to increase the frame rate. Column binning, where adjacent columns are combined, is also used. This reduces the horizontal resolution by a factor of two but increases the sensitivity in low light levels.

As a single chip camera, a colour image is obtained through a Bayer pattern colour filter array. Every pixel represents only one colour, and since only the red pixels are of interest, every second row is skipped. The green pixels are subtracted from the adjacent red pixel using a simple horizontal filter which produces an output every second clock cycle. This leaves an effective image resolution from the red pixels of 648×972resulting in a lateral resolution of approximately 1 mm, and a depth resolution of better than 1 mm. Due to the VGA native resolution a window of 640×480 was used for both the capture and display.

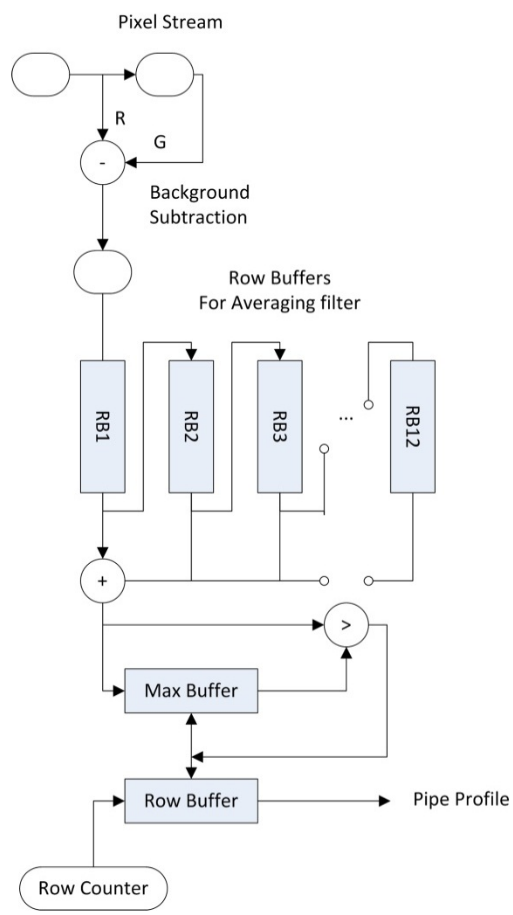

Figure 2. Initial logic for determining pipe profile

The next stage is to smooth the image vertically, and find the maximum value in each column. Instead of loading the whole image and processing each column individually, the parallel characteristics of the hardware can be exploited to process all columns in parallel as the data is streamed in. The smoothing filter used a 12×1 window which requires buffering the previous 12 rows. Instead of actually averaging the pixel values they are simply summed within the window. This effectively achieves the same result and removes the need to divide by 12.

The search for the maximum value in each column requires maintaining two additional memory buffers: one to store the maximum value in each column found so far; and one to store the corresponding row number. Once the first 12 rows have been smoothed, the maximum value buffer and row number buffers are initialized. For the subsequent windows the current value is compared to the current maximum for the column and if it is larger both the maximum value and row number are replaced. At the end of the image the maximum value buffer contains the maximum filtered red value in that column and the row number buffer holds the position of this. The values in these buffers can be used for comparing against a threshold and expected position to identify potential occlusions. The row number buffer contains a profile of the pipe.

The resulting images from the above steps are shown in Fig. 3. The images show the red component of the image, the green component along with the result from subtracting the green from the red. They also show the result after the averaging filter with the extracted profile overlaid in green.

The resulting profile was transmitted using RS232 which allows a maximum data frame of 8 bits. To account for the processing latency, the profile must be shifted left by two pixels, and up by 12.

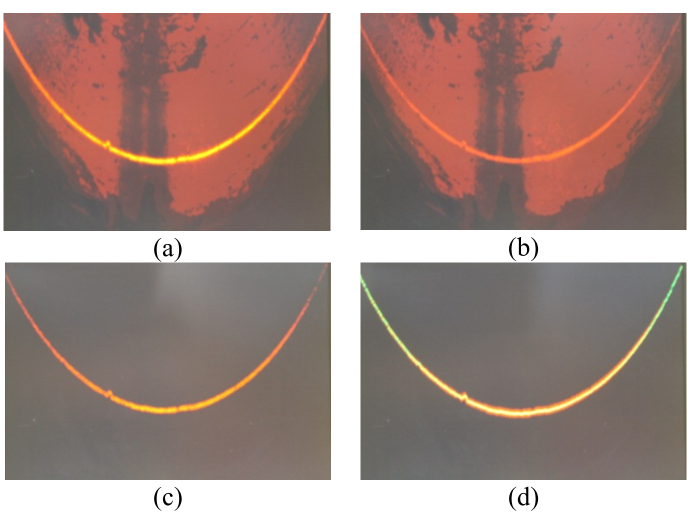

Figure 3. Steps of image processing; (a) shows the red channel of image, (b) the green channel, (c) shows the red minus the green, and (d) shows the results after filtering with the extracted profile overlaid in green.

B. *Final (DE0 Nano)*

This system was designed with the view in mind that the final implementation would be using a Terasic DE0 Nano. The DE0 was used for the development stages so to make use of its VGA output for debugging purposes.

When implementing on the DE0 Nano, the entire VGA component would be removed. The width of the camera window should also be extended to the full 648 pixels to give the largest possible field of view. The height could be reduced based on the radius of the pipe and the largest possible level of erosion. This would reduce the size of the image and potentially speed up the frame rate.

The final implementation also determines and flags potential occlusions. It then subtracts the profile from a known reference to give the level of erosion. The logic for this is shown in Fig. 4.

V. Considerations for Data Transmission

One main factor affecting the rate at which the robot can perform its inspection is the speed at which the profile data can be transmitted as this will directly affect either the speed or associated resolution. The resolution can be sacrificed to a certain extent; however, it is preferable to keep it as high as possible while still maximizing the speed.

A. *Data Compression*

There are two forms of compression that can be used to reduce the volume of data that needs to be transmitted, and hence increase the speed of transmission. Lossy compression reduces the volume of data at the expense of the colour, temporal, or spatial resolution. The compression that we are interested in at this point, however, is lossless compression. Here, all of the information is retained while the data volume is reduced. There are many different ways in which this can be done but one of the more appropriate ways in this case would be run length encoding. This encodes a value and length of

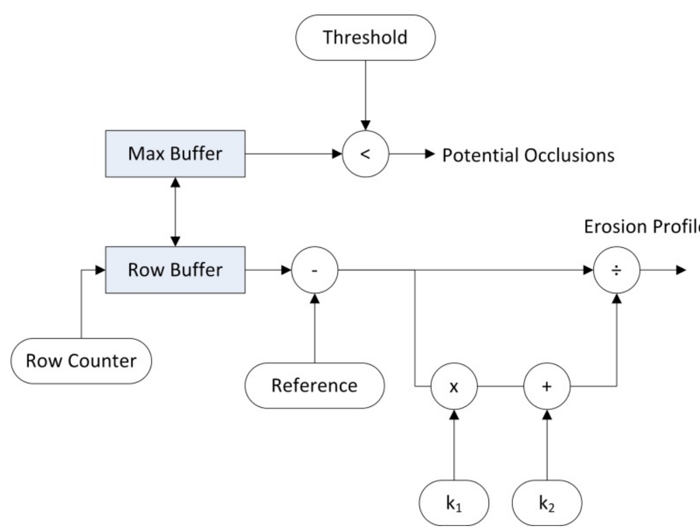

Figure 4. Additional logic for measuring erosion profile

consecutive data points with the same value and is particularly useful when there are long runs of the same data. This would be the case when encoding the erosion data as the erosion is likely to be even across the pipe. When there is no erosion the data should consist of long runs of zero which, once encoded, would dramatically reduce the data.

In a real world scenario, however, this would not be the case as there will always be a small amount of noise contained within the data. If the data is under a certain threshold it could be considered noise and reduced to zero. This is a form of lossy compression and may reduce the sensitivity of fine detail detection.

To further reduce the data a form of entropy encoding (such as Huffman coding) can be used. This assigns a code to every value and only the code is transmitted. Values that appear more frequently are assigned shorter codes but this requires the use of a lookup table which must either be transmitted or predetermined so that it can also be used to decode the data. This technique would be more effective when there is larger erosion on average throughout the pipe.

B. Wireless

There are many different wireless modules that could be used for wirelessly transmitting the profile data. These operate on a range of different radio frequencies and different protocols. Each of the different modules would impose limitations on the system in terms of speed, bandwidth and operation.

Wireless would not normally be an option for this application as the concrete walls of the pipe and the fact that the pipes are buried below ground would prevent the penetration of the radio signal. However, because only straight pipes are considered a wireless module on the robot could relay the data to a PC via a wireless router module at the entrance of the pipe. A wireless module such as Xbee would be ideal for this type of networking as they would offer a variety of options with differing ranges. They also have low power modules which are ideal for embedded solutions.

The wireless module used will have an effect on the speed at which the robot can travel. This comes from both the interface (which is usually serial such as RS232) and the frequency of the module. The baud rate is the governing factor in RS232. A higher baud rate means quicker data transfers but lower data integrity. That is, with a higher baud rate there are higher chances that data will be lost or corrupted during transfer.

The most common RS232 based Xbee modules have transfer rates of up to 250 Kbps. This would allow for up to 32 frames to be transmitted every second without any compression implemented. With a 10 mm resolution along the pipe this would be sufficient for the robot to be inspecting the pipe at a rate of 0.3 ms^{-1} worst case scenario. If this had not been the case there would have been a tradeoff between speed and resolution of the inspection.

VI. Testing and Results

A section of uniform pipe was used to test this system. The robotic platform was placed in the pipe and the profile data was transferred to a PC via the RS232 port on the FPGA. This data is shown in Fig. 5, and Fig. 6, along with the expected results calculated from the method described in [8]. Fig. 5, shows the entire image encompassed within the bounding box while Fig. 6, shows a close up of the relevant area. This shows that the results are fairly close to what was expected for the uneroded section of pipe. However, they are not close enough to be able to use this curve as the reference to subtract the raw data from in order to obtain the erosion profile. To be able to do this, a calibration process must be performed in order to match the curve to the actual results more accurately. The development of the required calibration along with the corresponding implications on the reference profile and erosion mapping is presented in [8] along with the results of this. The calibration would be done offline to generate an accurate reference which could then be downloaded onto the FPGA to be used in real time.

VII. Discussion and Conclusions

The suggested approach only provides the erosion profile of a cross section of pipe. In order to build a map of the entire pipe multiple cross sections must be inspected and stacked together. This requires knowing the position along the pipe of each cross section which could be acquired using a rotational encoder on one or more wheels. In order to increase the

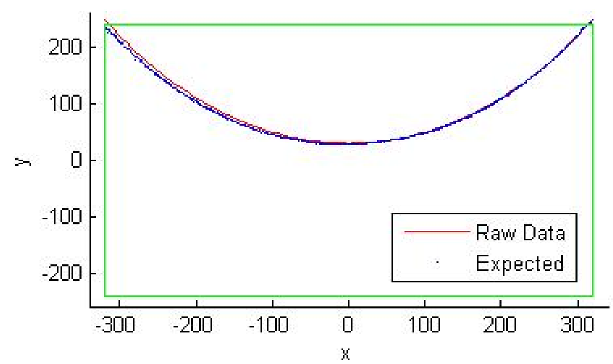

Figure 5. Raw data with expected data - whole image

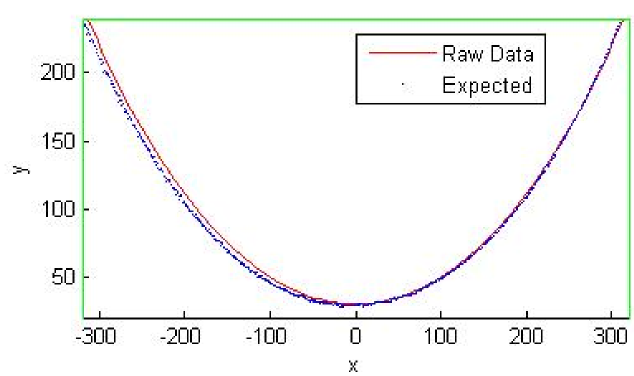

Figure 6. Raw data with expected data - relevant range

accuracy of this, multiple encoders could be used on non-driving wheels and the outputs compared to identify regions of slippage etc. This could then be used to trigger the imaging system in order to obtain regularly spaced samples.

Implementing the image processing on an FPGA has many advantages. Not only does the FPGA have more resources available than a standard embedded microcontroller but its parallelism characteristics can be exploited in order to maximize processing speeds. FPGAs are also low power devices making them ideal for battery operated systems. By embedding the FPGA in the robot the processing can be performed in real time close to the source which reduces the propagation of excessive data by extracting and transmitting only the relevant information. This also allows the data to be transmitted wirelessly removing the need for an umbilical cable to transfer the data. Appropriate compression techniques such as run length encoding could also be applied to further increase the speed of the transmission and, consequently, the overall inspection.

This paper has presented the development and implementation of an imaging system that can be used for high speed inspection of pipes. The results of this implementation have been presented and show the need for calibration for determining a reference curve to be used for calculating the level of erosion.

REFERENCES

[1] O. Duran, K. Althoefer, and L.D. Seneviratne, "Experiments using a laser-based transducer and automated analysis techniques for pipe inspection," Proceedings of the 2003 IEEE International Conference on Robotics & Automation (ICRA). Taiwan, vol. 2, pp. 2561-2566, September 2003.

[2] S.K. Sinha, S.R Iyer, and M.C Bhardwaj, "Non-contact ultrasonic sensor and state-of-the-art camera for automated pipe inspection," Proceedings of the 2003 IEEE Conference on Sensors. Vol. 1, pp. 493-498, October 2003.

[3] C. Ekes, and B. Neducza, "Pipe condition assessments using pipe penetrating radar," 14th International Conference on Ground Penetrating Radar (GPR). China, pp. 840-843, June 2012.

[4] O. Duran, K. Althoefer, and L.D. Seneviratne, "State of the art in sensor technologies for sewer inspection," IEEE Sensors Journal, vol.2, pp. 73-81, April 2002.

[5] D.G. Bailey, M. Jones, and L. Tang, "Real time vision for measuring pipe erosion," Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA). New Zealand, pp. 486-491, December 2011.

[6] Terasic, DE0 User Manual vol. Version 1.4: Terasic Technologies, 2009.

[7] Terasic, TRDB-D5M 5 Mega Pixel Digital Camera Development Kit vol. Version 1.2: Terasic Technologies, 2010.

[8] M. Jones, D.G. Bailey, and L. Tang, "Vision system and calibration for pipe inspection," Proceedings of the 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). New Zealand, November 2012.

[9] D.G. Bailey, "Design for embedded image processing on FPGAs," Wiley-IEEE Press, 2011.

[10] M.P. Battaglia, "Parrallelism for imaging applications," Northcon/93 Conference Record, pp. 52-56,October 1993.