Development of an Image Fringe Zero Selection System for Structuring Elements with Stereo Vision Disparity Measurements

Josef E. Grindley, Andrew J. Tickle and Lin Jiang

School of Electrical Engineering, Electronics and Computer Science, University of Liverpool, Liverpool L69 3GJ, United Kingdom

Email: joegrindley@hotmail.com, production_leader@hotmail.com

Abstract. When performing image operations involving Structuring Element (SE) and many transforms it is required that the outside of the image be padded with zeros or ones depending on the operation. This paper details how this can be achieved with simulated hardware using DSP Builder in MATLAB with the intention of migrating the design to HDL (Hardware Description Language) and implemented on an FPGA (Field Programmable Gate Array). The design takes few resources and does not require extra memory to account for the change in size of the output image.

1. Introduction

Image transforms that utilize a Structuring Element (SE) must be applied to an image that has been padded with zeroes or ones outside its fringes, whether zeros or ones are used is dependent on the image transform in use, however in this implementation only zeros are used. When implementing this in DSP, a separate SE consisting of a 5x5 8-bit AND gate array was used prior to the image transform, which was used to apply the packing bits. In this implementation a 5x5 array was used, as the image transform used was 5x5. However, should a 3x3 SE be needed then the data can handle sampling from the center 3x3 of the 5x5 array.

The are several benefits of using an Field Programmable Gate Array (FPGA) over a CPU when performing image processing operations in real time such as pipelining and comparatively lower power consumption, inherently any image processing functions used must be described in HDL. Many equivalent array padding functions are available, for example the function “padarray” in MATLAB performs the same operation in software as the proposed circuit and will be used as a comparison for the results section. One major advantage of using this implementation is that padding bits are applied around an array without the need to change the array dimensions hence the buffers used do not need to be adjusted in size. While applying the padding bits “on the fly”, only one clock delay is needed in the process.

2. Implementation in DSP Builder

The circuit works by using a SE consisting of an array of 8-bit AND gates, as shown in Figure 1, which allows zero to be selected in each appropriate element when the SE is centered at the fringes of

Andrew Tickle is now a Lecturer in Electrical Engineering at Coventry University

1
the image. This is done by using a bit pattern that characterizes the edges of the fringes, as illustrated in Table 1. As the center of the SE is always selected, the AND gate in the middle of the array is connected to an 8 bit integer “255”, although this could be done by connecting the center input to its respective output the AND gate was left in to eliminate synchronization issues and propagation delays. The bit pattern was input into the multiplexers as shown in Figure 2 where its input line to the multiplexer relates the number to the top left of each bit pattern in Table 3. The three numbers to the right are the decimal representation of each shaded section respectively: they are 8-bit making up the 24-bit needed for fringe pattern selection in the AND gate array. These were selected by the pattern select array illustrated in Table 1. The pattern select array was generated in MATLAB and input in parallel with the image. It was the same dimension as the input image and determined by the number to the top left of each section in Table 3.

Figure 1 – Illustration of Structuring Element using 5x5 AND gate array

The input to each AND gate from the pattern selection is 1-bit which is connected to all 8 AND gate inputs in parallel. This allows each element in the array to be selected at once using 1-bit as shown in Figure 5 but omitted in Figure 1 to simplify the diagram. The multiplexers shown in Figure 2 were configured as a look up table to select the correct pattern according to the input from the pattern select array. Table 1 shows the pattern select array which is of identical dimensions to the input image and used to select the pattern to be used at the part of the fringe that the AND gate array is centered over, hence applying the padding bits. The pattern select array is 5 bit and requires less memory than the input image. Table 2 shows a binary image for diagram simplicity, the circuit shown in Figure 7 is 8-bit and subsequent testing was done on an 8-bit grayscale image. With the appropriate modification, any bit length could be used. Table 3 shows each possible fringe pattern giving 25 possibilities, each pattern is separated into three 8-bit sections shaded appropriately for illustration. Each sections respective decimal value is in the cells to the right of each pattern; these are labeled L1, L2 and L3 which make up the input constants used by the multiplexers as illustrated in Figure 2. They are indexed and selected by the number to the top left of the pattern which can be seen in the pattern select array.
It should be noted that three multiplexers were used as DSP Builder would not allow multiplexer inputs that have a bus larger than 8 bits wide, hence three of them were needed to select the 24 bits required.

![Figure 2 - Illustration of Multiplexer configured to do pattern selection operation](image)

| 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 10 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |
| 11 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 |

![Table 1 - Illustration of pattern select array](image)

![Table 2 - Illustration of input image with packing bits](image)

3. Result out Images and Discussion

As the output image of Figure 7 shows packing bits that are a line two elements wide have been applied to the fringes of the input image in Figure 6. As the output of the AND gate array is equivalent to a 5x5 structuring element, the outputs were sampled and compared with the image in Figure 7 to validate the results and found to be identical thus demonstrating the design works correctly. When noise is apparent in the image input the noise is propagated to the output. The design does not do any noise filtering but the padding mechanism still works effectively regardless.
Table 3 – Illustration of each possible fringe position.

Figure 4 - DSP multiplexer pattern selection circuit
The results indicate the design is working correctly. The pattern select array was generated by the MATLAB m file as DSP builder was being used, when implemented on an FPGA the pattern select array must be generated either by a Nios processor or another CPU on the FPGA if the image frame size is variable, if not it could be loaded into memory as a constant when programming the FPGA. The system was originally designed to apply padding bits to the outside of two stereo images then apply the census transform where it was also found to function without any issues, this is intended for...
use in disparity distance measurements of which work is currently being undertaken for use in mobile robotics.

5. References

[1] Altera 2010 *DSP Builder Reference Manual 9.1 SP1*, www.altera.co.jp/literature/manual/mnl_dsp_builder.pdf (accessed 3 April 2011)

[2] *Manual 9.1 SP1*, www.altera.co.jp/literature/manual/mnl_dsp_builder.pdf (accessed 3 April 2011)

[3] Tickle A J, Smith J S and Wu Q H 2007 *Development of Morphological Operators for Field Programmable Gate Arrays* Proc. IOP Conference: Sensors and their Applications XIV, Liverpool John Moores University, Liverpool, UK

[4] MATLAB reference library, http://www.mathworks.com/help/toolbox/images/ref/padarray.html (accessed 7 March 2011)

[5] Kuhn M , Moser S and Islerl O 2004 *Efficient ASIC Implementation of a Real-Time Depth Mapping Stereo Vision System*, Department of Information Technology and Electrical Engineering, ETH Zurich

[6] Middlebury Computer Vision, vision.middlebury.edu/ (accessed 13 March 2011)