Design of Heterogeneous Environment

Jared Bold

Department of Electrical Engineering

Graduate Research Paper

Rochester Institute of Technology

Contents

List of Figures

List of Tables

1. Abstract

2. Introduction

3. Background

3.1 Hardware

The hardware platform used is the Xilinx ZC702 Evaluation Board, featuring a Zynq® XC7Z020 System on a Chip (SoC).

3.2 Operating System

The operating system selected to run on the ZC702 is Linux. The version used is maintained by Xilinx and is on kernel veresion 3.14.

3.3 Bilinear Interpolation

Bilinear interpolation (BI) is the process by which an unknown value is interpolated based on a 2 dimensional interpolation process using 4 known values. In its application in image processing, BI is used to scale images up and down. Unlike nearest-neighbor interpolation, which relies on only one known value resulting in blocky images, BI has creates smoother images when scaling due to its estimation of intermediate values.

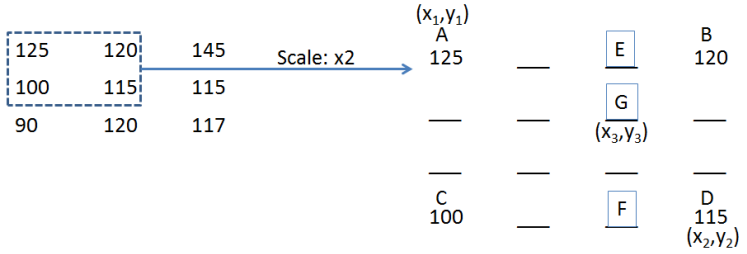

Figure 3.1: Bilinear Interpolation of Grayscale Image

The process of bilinear interpolation begins by expanding the known pixels to their scaled locations. This expansion process generates a window of unknown pixels, bounded by four corners of known pixels, as shown in 3.1. In order to calculate the the unknown pixel values, three equations must solved. To generate an final equation for the value, the variables shown in 3.1 will be used. The first equation is the horizontal interpolation of the value at E.

$$E = \frac{x_3 - x_1}{x_2 - x_1} B + \frac{x_2 - x_3}{x_2 - x_1} A \tag{3.1}$$

Next, the horizontal interpolation of the value at F is calculated.

$$F = \frac{x_3 - x_1}{x_2 - x_1} D + \frac{x_2 - x_3}{x_2 - x_1} C \tag{3.2}$$

Finally, the vertical interpolation of value at G is calculated.

$$G = \frac{y_3 - y_1}{y_2 - y_1} F + \frac{y_2 - y_3}{y_2 - y_1} E \tag{3.3}$$

$$\alpha = x_3 - x_1, \beta = x_2 - x_3, \gamma = y_3 - y_1, \omega = y_2 - y_3 \tag{3.4}$$

By substituting (3.1) and (3.2) into (3.3) and allowing variables to be renamed as in (3.4), the BI equation for a pixel is determined.

$$G = \frac{1}{(\omega + \gamma)(\beta + \alpha)}[\gamma(\alpha D + \beta C) + \omega(\alpha B + \beta A)] \tag{3.5}$$

By applying (3.5) to every unknown pixel in every window that is generated from the scaling process, an image is enlarged or shrunk with minimal loss of quality. This technique is easily extended to multiple color channel images such RGB by operating on each colorplane individually and ensuring that pixel offsets are correctly calculated.

4. Design

4.1 Common Design Components

4.1.1 Tagged Image File Format

All images used throughout this research are Tagged Image File Format (TIFF) images. TIFF images consist of an image header that describes the byte order, TIFF version number, and offset to the image file directory[2]. The image file directory stores the offsets to the directory entries and varies in size based on the number of images contained within the TIFF image. The directory entry is then broken down into several sections including a Tag section which consists of a number of tags which give information about the image including the bits per pixel, compression scheme, image length and height, and many more. The directory entry is lastly completed with the offset to the image data. Figure 4.1 shows a more indepth representation of the TIFF structure for multiple image TIFFs.

For the purposes of this research, only uncompressed TIFF images are used in order to eliminate the need for resource intensive decompression and compression of input and output images. To support the reading and writing

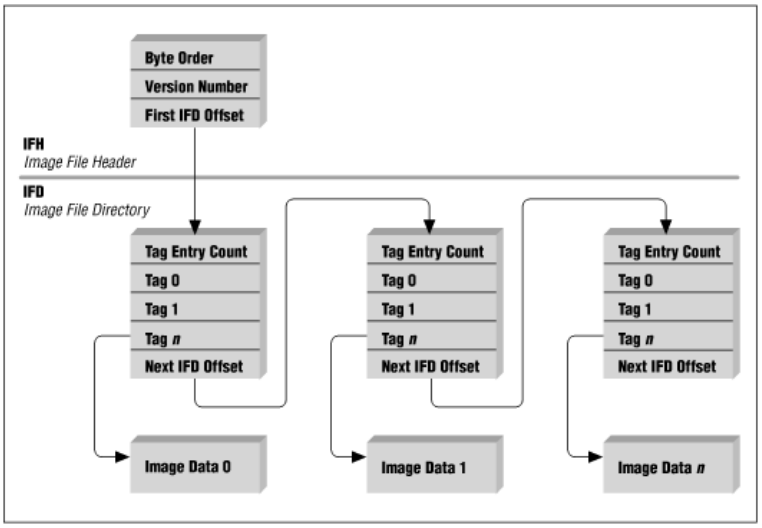

Figure 4.1: TIFF File Structure[1]

of these image files, the LibTiff libary is cross-compiled for use on the ZC702's PetaLinux OS.

5. Results

6. Conclusion

Bibliography

[1] James D. Murray and William Van Ryper. *Encyclopedia of Graphics File Formats, 2nd Edition.* O'Reilly Media, 1996.

[2] Richard H. Wiggins, H. Christian Davidson, H. Ric Harnsberger, Jason R Lauman, and Patricia A. Goede. Image file formats: Past, present, and future. *RadioGraphics*, 21:789–798, 2000.