Binary Image Converter Using Verilog (January 2019)

Von Errol L. Ang, Dann Paulo P. Javierto, Nolasco Elezar S. and Lanz Clinton D. Valdez

***Abstract*— It is important in image processing to extract objects from their background into a binary image. Binary image is used as input to feature extraction process and have an important role in generating unique feature to distinguish several classes in pattern recognition. Image processing in Verilog has been popular, but Verilog has limitations in reading images from a file. The limitations are the size of the image is fixed and the only input is a hex type that would output a bmp type. A binary image which is one of the techniques in image processing is being relied on other software before inputting the image in Verilog to process it. This paper proposes an image processing algorithm in Verilog to convert an image to binary image.**

***Keywords*—Verilog, binary image, algorithm, image processing, bmp, software.**

I. INTRODUCTION

THIS project aims to show and process an image in png format into a binary image in bitmap format in Verilog. First of all, Verilog cannot read images directly. To read the image file in Verilog, the image is required to be converted from png format to hexadecimal format, this is implemented through Python. After producing a hex file in Python, Verilog uses a command read whether the image data is in a binary text file. After reading the image, hex file, the RGB image data are saved into memory and read out for processing. Some simple operations are implemented in Verilog such as inversion, brightness control, and threshold operations. In this case, the threshold operation was used as a reference to create the necessary image processing to produce the binary image of the input hex file. The following are the objectives of the project; is to create a Verilog program that will convert an image into binary image, to output the binary image's bmp file to see the resulting binary image, and to compare the time it takes to output a binary image if the file format of the original image was from a bmp or png file.

II. REVIEW RELATED LITERATURE

This presents the Review of Related Literature from foreign sources. This includes the gap, method, and results presented and defined conceptually and operationally for clarity. In the image processing, image binarization is used as a general tool for image segmentation of discriminating objects from background in various applications such as automatic target tracking, object recognition, image compression, image analysis, and object separation. Some recent researches i.e. need a perfect binary image to separate objects before doing classification and also need a perfect binary image to extract wood fiber from background. Because of its important role, many researches of image binarization have been developed from Otsu algorithm and appear some image segmentation techniques as presented in. This paper is a continuation of and to distinguish several classes of wood in object recognition but it's still having difficulty to get a perfect binary image. This maybe one reason why still has not reached maximum performance. This paper proposes an image processing method to convert RGB image into binary. Statistic-parameter is a value that is used to determine the number of images as segmentation result. We use the mean and standard deviation as a reference for determining statistic-parameter. The next process is finding first limit and end limit. First limit is 1-dimensional matrix that contains the first index of the selected range at histogram. While end limit is 1-dimensional matrix that contains the end index of the selected range at histogram. This selected range is obtained from matrix that contains pixel 1 and represents the pixel intensity whose occurrence is greater than the specified statistic-parameter. From Figure 3, it is obtained first limit = [0 3] and end limit = [1 3]. It means that there are 2 selected ranges at histogram, i.e. from 0 to 1 and from 3 to 3 The algorithm still generates more than one binary image is the weakness of this research. However, each sample has one of the results that contains the desired object. It is important in next research to automatically select only one binary image that contains the desired object. That is why this algorithm can be developed and can be used to convert RGB image to binary as needed.

III. METHODOLOGY

For the Verilog program to read images, it requires the image to be converted into a readable hex file. In this case, a Python program was used in order to do the job for this. Two separate py files were created to separately convert an image that is in BMP and an image that is in PNG to hex..

*tohex.py - C:\Users\Paul\Downloads\...

File Edit Format Run Options Window Help

```
import cv2

img = cv2.imread('input.png')
print(img.shape)

cy=img.shape[0]
cx=img.shape[1]
f = open('input.hex', 'w')
for row in img:
    for col in row:
        str="%02x"%(col[0])
        f.write(str)
        nw = "\n\r"
        f.write(nw)
        str="%02x"%(col[1])
        f.write(str)
        str="%02x"%(col[2])
        nw = "\n\r"
        f.write(nw)
        f.write(str)
        nw = "\n\r"
        f.write(nw)
f.close()
```

Ln: 24 Col: 0

Source code of the Python Program for the PNG to hex

bmptohex.py - C:/Users/Paul/Downloa...

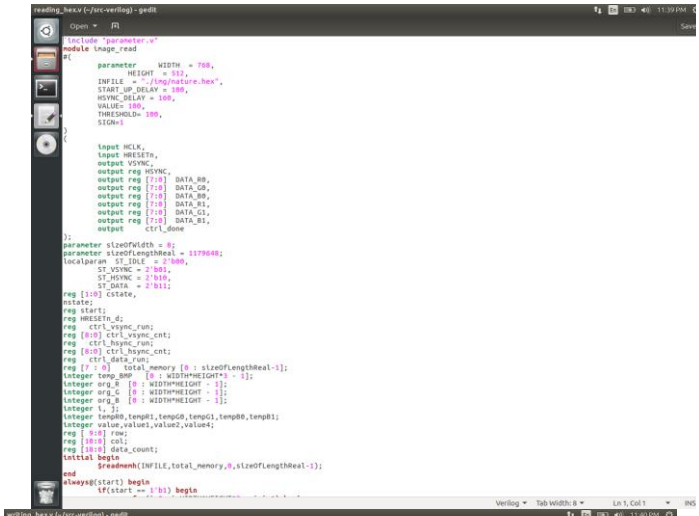
File Edit Format Run Options Window Help

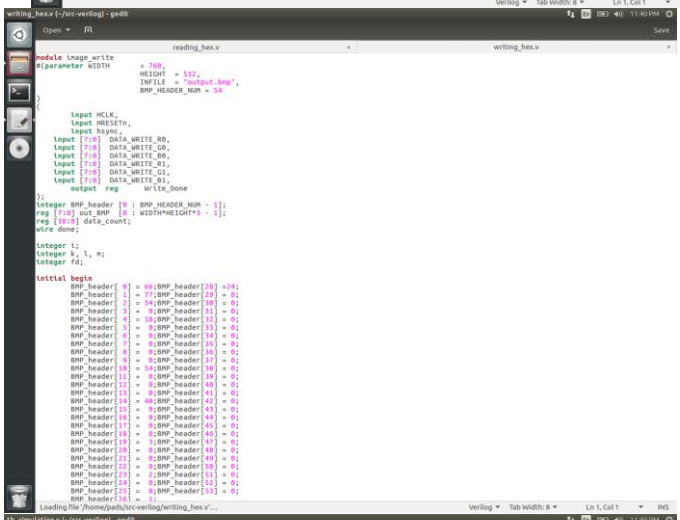
```
import cv2

img = cv2.imread('input.bmp')
print(img.shape)

cy=img.shape[0]
cx=img.shape[1]
f = open('input.hex', 'w')
for row in img:
    for col in row:
        str="%02x"%(col[0])
        f.write(str)
        nw = "\n\r"
        f.write(nw)
        str="%02x"%(col[1])
        f.write(str)
        str="%02x"%(col[2])
        nw = "\n\r"
        f.write(nw)
        f.write(str)
        nw = "\n\r"
        f.write(nw)
f.close()
```

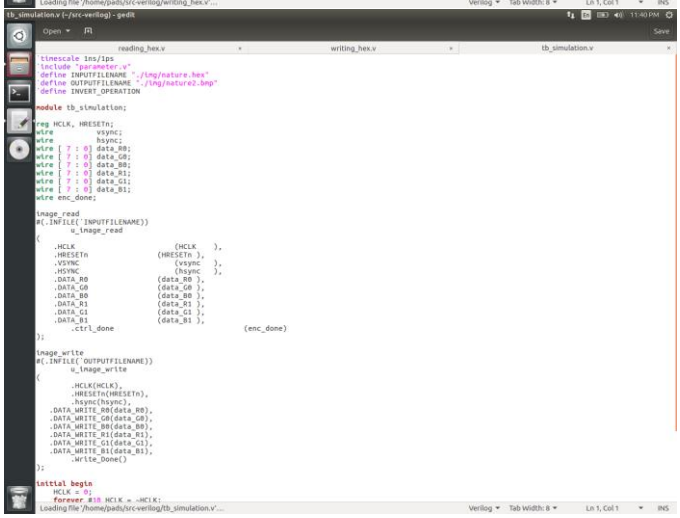
Ln: 3 Col: 27

Source code of the Python Program for the BMP to hex

After that, the Verilog program will read the processed hex file with the help of a header file called parameter.v which is commonly used in this type of applications which handles the maximum size of the image and its path. The program will then read the content of the hex file line by line and will try to read its colors and process it to produce the binary image. The processing of the image involves looking at its colors and if the colors are bright, it will be white otherwise a black which is done pixel by pixel of the image. After that, the processed image file will be written in a bmp file just to display the output of the image.







IV. TESTING TABLES

Now that the Verilog program is working, we tested the image nature which is in two file formats, BMP and PNG and was processed into a hex file using Python. The hex file produced by the python program was processed in the Verilog code. We compared how fast will the program process these two hex files from different file formats separately and ran a timer to measure it while doing the test.

Trial Number	Time in BMP	Time in PNG
Trial 1	168 s	160 s
Trial 2	147 s	139 s
Trial 3	178 s	172 s

V. CONCLUSION

We were able to make a program in Verilog that can convert a binary image by reading the hex file of an image where a Python program was used to convert an image to a hex. The Verilog program was successful in producing the binary image of its input hex file which would be very useful for Image Processing using Verilog exist nowadays and Pre-Processing an image to convert it into binary image can be directly done by Verilog.

In our test, it appears that Verilog handles images that is in PNG file format, faster than the BMP when converted into hex. Therefore, it is recommended to use the PNG file format when image processing in Verilog especially when multiple images are being processed.

REFERENCES

[1] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979; 9(1): 62-66.

[2] Hetal J Vala, Astha Baxi. A Review on Otsu Image Segmentation Algorithm. International Journal of Advanced Research in Computer Engineering & Technology. 2013; 2(2): 387-389.

[3] Ratri Dwi Atmaja, et al. The Detection of Straight and Slant Wood Fiber Through slop angle fiber feature. TELKOMNIKA Indonesian Journal of Electrical Engineering. 2015; 14(2): 318-322.

[4] Kwang Baek Kim. Image Binarization using Intensity Range of Grayscale Images. International Journal of Multimedia and Ubiquitous Engineering. 2015; 10(7): 139-144.

[5] Allan G Bluman. Seventh Edition of Elementary Statistics: A Step by Step Approach. McGraw-Hill. 2009.

[6] Ratri Dwi Atmaja. Wood image real-time segmentation algorithm based on video processing. International Journal of Imaging and Robotics. 2015; 15(1): 12-18.

[7] Ratri Dwi Atmaja, et al. One Layer Object Separation Algorithm in Binary Image. Indonesian Journal of Electrical Engineering and Computer Science. 2016; 1(1): 215-220.