MEDINFO 2013
C.U. Lehmann et al. (Eds.)

© 2013 IMIA and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-289-9-987

CUDA Implementation of Histogram Stretching Function for Improving X-ray Image

Yong H. Lee [a], Kwan W. Kim [a], Soon S. Kim [a]

[a] *Department of Computer Engineering, Halla University, Wonju, Gangwon-Do, Korea*

Abstract and Objective

This paper presents a method to improve the contrast of digital X-ray image using CUDA program on a GPU. The histogram is commonly used to get the statistical distribution of the contrast in image processing. To increase the visibility of the image in real time, we use the histogram stretching function. It is difficult to implement the function on a GPU because the CUDA program is due to handle the complex process to transfer the source data and the processed results between the memory of GPU and the host system. As a result, we show to operate the histogram stretching function quickly on GPU by the CUDA program.

Keywords: GPU, CUDA, X-ray image, Histogram

Introduction

A histogram means the cumulative probability density of data. In a medical image, an image histogram value can be used to set the threshold value for edge detection, segmentation of image such as X-ray, MRI. It cumulates the number of pixels for each tonal scale. The horizontal axis of the histogram represents the total range of the tonal scale, while the vertical axis plots the number of pixels in that particular scale value in the image. The left side of the horizontal axis shows the dark areas, and the right side shows the white areas. The histogram for a dark image has the majority of its pixels on the left side, the histogram for the image with white contrast has most of its pixels on the right side in the histogram, relatively. In these cases, the dynamic range of the image is reduced. It is difficult to get the useful clinical information from the image which is shifted the histogram to one side of left or right. Recently, the study has been tried to improve the contrast for these image in real time. We implement the CUDA program on GPU to increase the dynamic range of the image's histogram using the histogram stretching function.

Methods

Histogram Stretching Function

Stretching a histogram is a method to extend the dynamic range of the image's contrast. This method usually increases the global contrast and visibility of images, especially when the useful information in an image concentrates in specific contrast values. The histogram stretching function is calculated as

$$\mathrm{OutImg}[x][y] = \frac{\mathrm{InImg}[x][y] - \mathrm{Low}}{\mathrm{High} - \mathrm{Low}} \times 2^{B} \quad (1)$$

where *InImg[x][y]* is a source image, and *OutImg[x][y]* is the image with a stretched contrast. And *B* is the coded bits of an image. *Low* refers to the lowest intensity value of the image, and *High* is the highest value, relatively. The stretching function is useful to improve the images with backgrounds and foregrounds which are both bright or both dark. Especially, the method shows to better performance in X-ray images with bone structure.

CUDA Implementation

We use the CUDA toolkit ver. 4.2.9, Visual C++ ver. 2010, and GPU SDK ver. 4.2.9 as the development tools. And the test image uses 8-bit image which converts the 14-bit images to the 8-bit in an X-ray image. The CUDA program is as follows.

```
//    Histogram Stretching //
__global__                                    // Kernel  function
void UpHisto_Cal(int* Input, int* HistoGram)
{
int tidx, bidx;
     tidx = threadIdx.x;                      // Thread x value
     bidx = blockIdx.x;                       // Thread y value
     HistoGram[Input[bidx*256 + tidx]]++;// Thread x,y array
}
                :
cudaMemcpy(d_Input, Input, sizeof(int) * 256 * 256,
          cudaMemcpyHostToDevice);
cudaMemcpy(d_HistoGram, HistoGram, sizeof(int) * 256,
          cudaMemcpyHostToDevice);  // Memory allocation
UpHisto_Cal <<<256, 256>>>(d_Input, d_HistoGram);//Kernel call
cudaMemcpy(HistoGram, d_HistoGram, sizeof(int) * 256,
          cudaMemcpyDeviceToHost); // Copy memory to Host from
          Kernel
                :
```

Results

As a result, we implemented the histogram stretching function by the CUDA program on a GPU. The Image with 8-bit grayscale used to test the function. Figure 1(a) is a source image and the histogram. Figure 1(b) shows the stretched histogram and the image corresponding to the histogram. The image with stretched histogram shows the improved visibility.

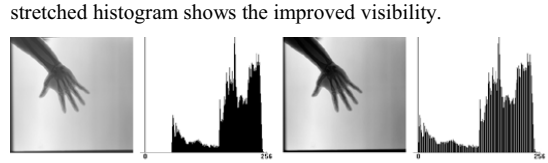

Figure 1- Source image and Stretched image
(a) source X-ray image and the histogram (b) stretched image and the histogram

Acknowledgements

This research was financially supported by the Wide Area Economic Belt Development Project for the Ganwon Leading Industry.