Modulation Transfer Function (MTF) Measurement on Cylindrical Object: ASTM-E1695-95

Manish Mahesh Dalvi, Likith Rana Vayala, and Kartik Nayak

Institut fuer parallele und verteilte Systeme, IPVS, University of Stuttgart
st164241@stud.uni-stuttgart.de, st164244@stud.uni-stuttgart.de,
st164247@stud.uni-stuttgart.de

Abstract. Modulation transfer function (MTF) determines how much contrast in the original object is maintained by the detector. In other words, it characterizes how faithfully the spatial frequency content of the object gets transferred to the image. We implement a method for calculating the modulation transfer function of a computed tomography (CT) system of a cylindrical object by pixel binning method and polynomial fitting technique. The MTF was measured from a circular image by use of circular edge method and pixel binning technique was applied to an edge response function (ERF) to eliminate noise because the edge method is very susceptible to noise in the ERF in a CT image. This technique provides MTF without any fluctuations that can be caused by the noise in the image.

1 Introduction

Modulation transfer function (MTF) is one the most widely used metric for spatial resolution assessment, and it is commonly used for the evaluation of spatial resolution properties of computed tomography (CT) systems [4]. Therefore, there is a strong demand for simple method of measurement of the MTF. In addition, it has become important to measure the MTF for a wide range of doses and object contrasts with increasing popularization of advanced CT systems.

Several studies have shown various edge methods, one of which is the MTF measured from disk images of a cylindrical object, that is based on ASTM standard. Although the circular edge method can be implemented easily, this method is affected by noise because of the derivative of edge response function (ERF) that yields point spread function (PSF) ; thus it is extremely important to eliminate the noise in the ERF.

In this implementation, three images of different resolution are compared with pixel binning technique, thus providing MTF calculation for even low contrast image that is affected by noise. To further boost up the calculation speed, the parallelism of graphic processor unit (GPU) has been explored, with implementation on both CPU and GPU, the algorithm calculation speeds have been compared and shown how GPU handles shallow compute intensive tasks better than CPU.

2 Cylindrical Edge Method

ASTM standard method for measurement of computed tomography system performance provides the method for calculating the modulation transfer function of cylindrical object. The test method provides instruction for determining the spatial resolution and contrast sensitivity in x-ray and gamma ray computed tomography images. The determination is based on examination of CT images of a uniform disk of material (shown in Fig. 1) and the spatial resolution is derived from an image analysis of the sharpness at the edge of the disk [1].

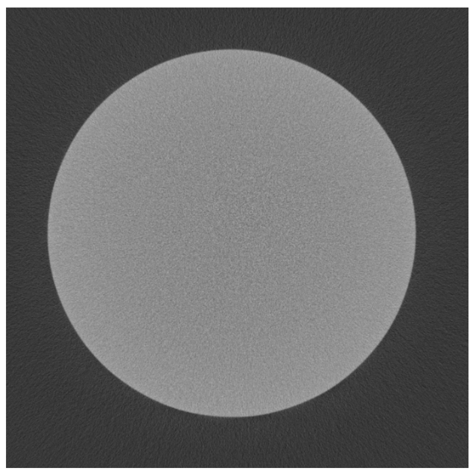

Fig. 1. Circular Edge Method

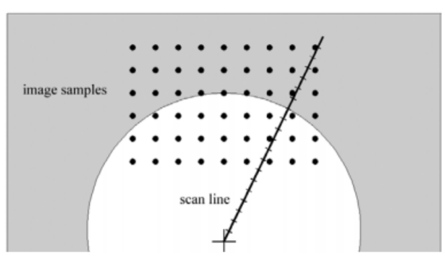

Fig. 2. Radial Scan Lines [2]

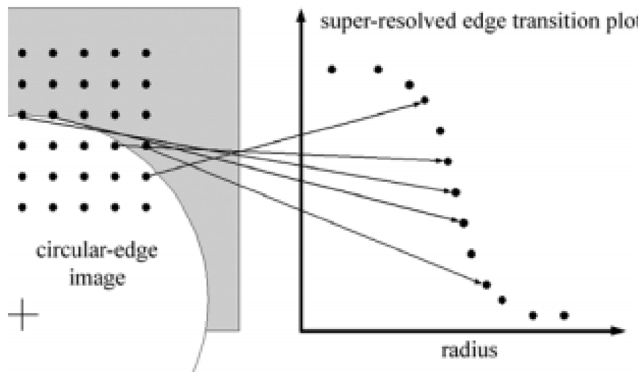

Fig. 3. Relationship between image values and edge transition curve [2]

The curves of edge response function, point spread function and modulation transfer function are shown in Fig. 4, Fig. 5 and Fig. 6 respectively.

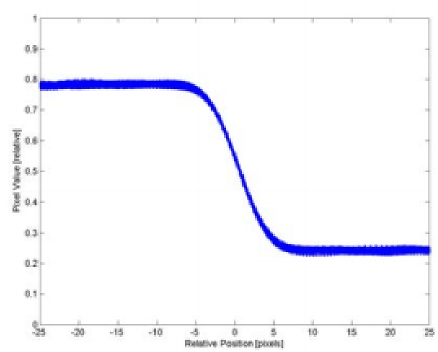

Fig. 4. Edge Response Function

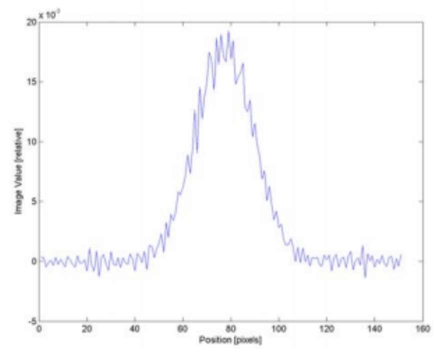

Fig. 5. Point Spread Function

The use of radial scan lines (shown in Fig. 2) sets the cylindrical edge method from slanted edge method. The first step in the process is to calculate position of the center of the test target. Radial scans are then performed to scan the edge positions. This is done by choosing the inner and outer circle with respect to center of a circle that bracket the circle and segregating the region between inner and outer radii with bins sized to a small fraction of a pixel which results in bins with pixels that are almost at the same distant from the center of the circle. Averaging the members of each bins will provide the table of values of constant increments from inner to outer radius. Once the entire table is iterated, starting at one end, the pixel values are smoothed by performing a piecewise, least-squares cubic fit which results in a polynomial and replacing the center value with that predicted by the polynomial which is a function of distance. The number of values to include in the fit should be large compared to the order of the polynomial and small compared to the fine ERF structure. The projection of pixel values on the ERF is shown in Fig. 3. For each fit, the analytical derivative of the resultant polynomial is calculated and its numerical value at the center of the piece-wise window is determined. The normalization of the resulting curve provides PSF. The magnitude of fourier transform of PSF is performed later to obtain the MTF of the CT image [1].

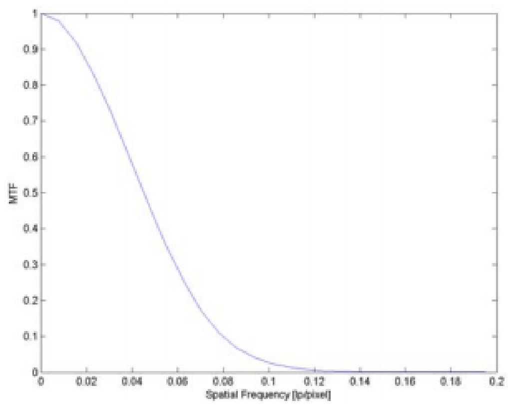

Fig. 6. Modulation Transfer Function

3 CPU vs GPU

CPU is known as brain for every ingrained system. CPU comprises the arithmetic logic unit (ALU) accustomed quickly to store the information and perform calculations and control unit (CU) for performing instruction sequencing as well as branching. CPU interacts with more computer components such as memory, input and output for performing instruction.

GPU is used to provide the images in computer games. GPU is faster than CPU and it emphasizes on high throughput. It is generally incorporated with electronic equipment for sharing RAM with electronic equipment that is nice for the foremost computing task. It contains more ALU units than CPU.

Fig. 7 shows the comparison of how building blocks are organized in CPU and GPU. A GPU can only do a fraction of the many operations a CPU does, but it does so with incredible speed. A GPU will use hundreds of cores to make time-sensitive calculations for thousands of pixels at a time, making it possible to display complex 3D graphics. However, as fast as a GPU can go, it can only really perform simpler operations.

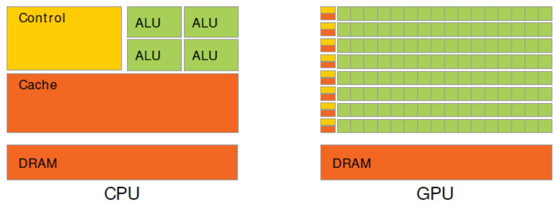

Fig. 7. Central Processing Unit vs Graphics Processing Unit [5]

4 Implementation

4.1 Implementation on CPU

For the calculation of the modulation transfer function, three test CT images were given, named input 4, input 7, input 10, which are of different resolution with pixel size 31.75um, 18.1432um and 11.99966um respectively. The resolution of the images in the reverse order to the pixel size of the given images. For any given input CT image, the circle was detected using Hough Transform, and the the radius of the given circle was detected. Once the given image was converted to gray scale, GuassianBlur() function was used reduce the noise to avoid the false detection of the circle. The inner radius and the outer radius with respect to the center of the circle are selected to comfortably bracket the edge. This is done by arbitrarily setting the region of interest radius (here 10), and adding and subtracting to the actual radius of the circle and forming two arbitrary circles to bracket the edge. The radius can be changed in order to consider more pixels bracketing the edge radius so as to obtain more information. The pixel is considered to be inside the region of interest if it satisfies

$$(r + 10)^2 \leq (x^2 + y^2) \leq (r - 10)^2$$

where r is the radius of the detected circle, 10 is the selected ROI bracket, x and y are the coordinates of the current pixel. This is the region of interest that is used to further calculate edge response function, point spread function and modulation transfer function.

Once the region of interest is selected, the region needs to be extracted. This is done by computing the distance to the center of mass for all pixels between inner and outer radii and a table of pixel values is generated. The table will contain pixel intensity values and the pixel distance values and is sorted in the order of their distance from the center of mass. The pixel values are then segregated into bins by gathering all pixels near the current pixel which are inside the range mentioned in the bin size which results in bins having pixels that are almost at the same distance from the center. This is done by setting the practical size for each bin. In the implementation, the bin size is set to 0.032, 0.0182857, 0.0128 for the given three images respectively as it eliminated most of the noise in the curve has been retained but this can also be changed in order to obtain different results. The member of each bins are averaged to obtain table of values at constant increments from inner to outer radii. The least square polynomial fit function is a cubic fit function that starts at one end of the table and iterates through the content of values till the end of the table. The content of values are processed thus performing a piece-wise, least squares cubic fit to produce a polynomial and replacing the center value of the fit with the value predicted by the polynomial which is a function of distance. It is recommended that the number of values to include the fit should be large compared to the order of the polynomial and small compared to the fine ERF structure [1]. We have used 25, 55, 79 as fit size for the three provided images respectively.

Plotting the values from the cubic fit provides the ERF for the implementation. Starting at one end of the table, and iterating through the table of values of ERF, the piece-wise, least squares cubic fit was performed to the ERF with the same number that were used to smooth the data. Each set of values generate a polynomial which is then differentiated and the resultant polynomial is used to calculate values as a function of distance. The values are normalized to fit the curve to unity to obtain the point spread function (PSF).

Fourier transform of the PSF results in modulation transfer function. In the CPU, the fast fourier transform (FFT) is calculated using a recursive function which is called from perform_fft() function and is used to obtain the timing performance. Once the complex array has been obtained, the magnitude of the complex number is calculated. Normalizing at zero frequency to unity produces the Modular Transfer Function (MTF).

4.2 Implementation on GPU

The algorithms used in the implementation of GPU is identical to the one used on the implementation of CPU. Since GPU provides parallel processing as compared to CPU, the execution of the algorithm takes lesser time and the implementation should provide result faster than the implementation on the CPU as GPU's can process the pixels in parallel fashion.

The center of the circle and its radius is detected on the CPU side. The pixels in the region of interest are projected along the direction of the estimated circular edge which is implemented on the GPU side since parallel computation of the distance of each pixel from the center of circle is possible as center of the circle is already known. Sorting and data binning technique functions are implemented on CPU in order to reduce the complexity. In pixel-binning technique, pixels which lie at almost same distance from the center are put into one bin and the distance is set initially as bin size. This process is iterated over all pixels resulting in almost equal sized bins. This process is particularly useful to smooth the ERF curve. The pixel values of each bin are then averaged and the resultant pixels are plotted to obtain the super-sampled ERF.

The ERF is differentiated to obtain PSF. Magnitude of the Fast fourier transform of generated PSF is normalized to produce MTF curve which is made faster by GPU implementation.

5 Results

5.1 Curves

Fig. 8, Fig. 9 and Fig. 10 represent the edge response function, point spread function and modulation transfer function of the input 4 respectively. In the

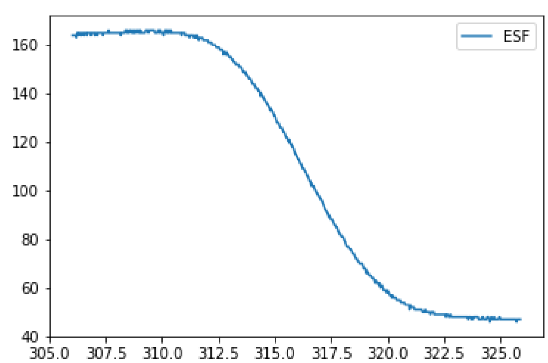

Fig. 8. Edge Response Function Curve for Input4

implementation, input 7 and input 10 have been tried out, where input 7 is higher resolution image compared to input 4 and input 10 is the most pixelated image of the three. It was also observed that MTF measured without pixel binning and cubic fit technique fluctuates irregularly, especially at higher spatial frequency, owing to the noise remaining in the averaged ERF, whereas the MTF with pixel binning and polynomial curve fitting indicates no fluctuations for the entire spatial frequency range.

5.2 Speed up obtained on GPU

Since the GPU is specifically designed for computing graphical displays, rendering images and shallow parallel processing, it must execute the calculation of

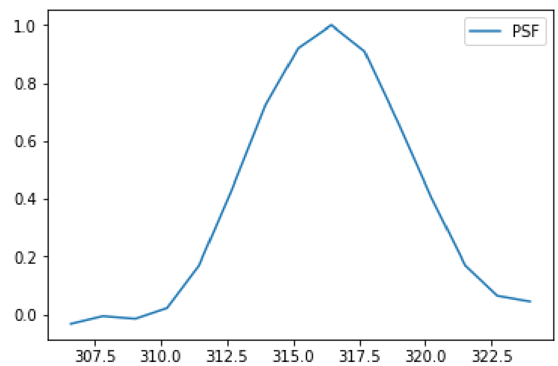

Fig. 9. Point Spread Function Curve for Input4

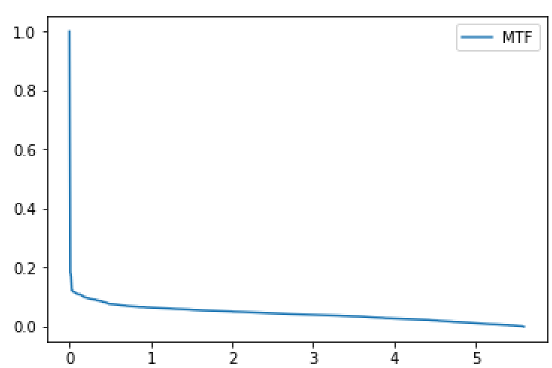

Fig. 10. Modulation Transfer Function Curve for Input 4

modulation transfer function at a much higher rate than CPU. In our results, we observed that CPU takes a processing time of 117.803ms for the calculation of MTF while GPU reduces the processing time significantly to 0.293ms. That is, the speed obtained on the GPU is approximately 400 times higher speed than on CPU. However, execution time comparison as a performance metric is done for only two functions - calculation of pixel values from the center and fast fourier transform.

6 Conclusion

MTF is one of the finest tools available to quantify overall imaging performance of a system in terms of resolution and contrast. Various methods have been used to calculate MTF, namely, knife edge, point source, pulse, periodic target etc., and every method has its own advantages and limitations. The cylindrical edge method used in this implementation is superior to slanted edge method for cameras with rotationally-symmetric distortion and blur. It is not an appropriate method for cameras that have more complicated aberrations, such as astigmatism [3].

Since the cylindrical edge method is computationally intensive compared to slanted edge method, it was practical to use slanted edge method to calculate MTF in the past. However, with modern computers being so powerful, with powerful CPU and GPU performance, the computation is within the performance capabilities. The same can be observed from the results. CPU performs the MTF based on cylindrical edge method in milliseconds, and GPU, using its many ALUs, does it in microseconds and gives almost 400 times the performance improvement over CPU. Thus for the calculation of MTF using cylindrical edge method, GPU will fare much better than CPU.

References

1. *Standard Test Method for Measurement of Computed Tomography (CT) System Performance1.* An American National Standard.
2. Richard L. Baer. The circular-edge spatial frequency response test. *Agilent Laboratories.*
3. C. C. Li M Mazumdar, J. K. Sihna. comparison of several estimates of edge point in noisy digital data across a step edge. *IEEE Computer Vision and Pattern Recognition Conference*, 1985.
4. Judy PF. The line spread function and modulation transfer function of a computed tomographic scanner. *Med Phys.*, 1976.
5. Kaicong Sun. *OpenCL Introduction.*