Implementation of Modulation Transfer Function (MTF) Measurement on Slanted Edge: ISO 12233:2017

AYAZAHMED KAZI, GUNJAN GUPTA

Institut für Parallele und Verteilte Systeme, IPVS, University of Stuttgart
st164267@stud.uni-stuttgart.de, st164236@stud.uni-stuttgart.de

Abstract. Modulation transfer function (MTF) is a commonly used measure to compare the performance of optical systems. While multiple techniques can be employed to estimate MTF, the preferred routine for computing the spatial resolution performance of photographic equipment and X-ray CT systems is the slanted edge method. As specified in ISO standard 12233, this method calculates MTF as a function of the vertical spatial frequencies by analyzing a user-defined rectangular region of interest (ROI) in an image of an extracted slanted-edge target. Next, pixel-binning technique is applied to plot the Edge Spread Function (ESF). Lastly, this ESF is differentiated and Fourier-transformed to estimate MTF. To achieve such high data parallelism in CPU, the speedup factor gets limited to the number of cores. With a motive to accelerate execution time, this report outlines the performance of slanted-edge method to estimate MTF between GPU and CPU.

1 INTRODUCTION

To properly define MTF, it is necessary to introduce terms that truly characterize image performance: spatial frequency, resolution and contrast.

Spatial frequency is a characteristic of any structure that is periodic across position in space. It is a measure of how often sinusoidal components (as determined by the Fourier transform) of the structure repeat per unit of distance.

Resolution refers to imaging system's ability to distinguish object detail. It is often expressed in terms of line-pairs per millimeter (lp/mm) (where a line-pair is a sequence of one black line and one white line and has a contrast of unity). High-resolution images are those which exhibit a large amount of detail as a result of minimal blurring. Conversely, low-resolution images lack fine detail.

Contrast or modulation is defined as the capability to distinguish between differences in intensity in an image. As the spatial frequency increases, the contrast of the image decreases. For the image to appear defined, black must be truly black and white truly white, with a minimal amount of grayscale in between.

Now that the components are defined, a consolidated definition of MTF can be established. It is a way to transfer contrast at a particular resolution from the object to the image. It is a plot of contrast, measured in percent, against spatial frequency measured in lp/mm. This graph is customarily normalized to a value of one at zero spatial frequency (all white or black). As spatial frequency increases on the test target, it becomes increasingly difficult for the lens to efficiently transfer this decrease in contrast, as a result, MTF decreases.

2 IMPORTANCE OF MTF

MTF is one of the best tools available to quantify the overall imaging performance of a system in terms of resolution and contrast. Every component within a system has an associated MTF and as a result, contributes to the overall MTF of the system. This includes the imaging lens, camera sensor, video cables and so on. The resulting MTF of the system is the product of all the MTF curves of its components. By analyzing the system MTF curve alone, it is sufficient to determine which combination will yield sufficient performance. This allows a system designer to make the appropriate selection to leverage maximum resolution of the image.

MTF is frequently used by scientists in both academia and industry to assess the contrast between features within images formed by an optical system. Following are the domains in which usage of MTF finds its maximum potential:

1. Spatial resolution of satellite borne cameras
2. In position sensitive detection and imaging
3. Sharpness of photographic imaging systems
4. Spatial resolution properties in Computed Tomography (CT) systems

In the past, various methods have been used to estimate MTF like knife edge, point source, pulse, periodic target; each with its own benefits and limitations. However, the most effective one that produces a good approximation of MTF till date remains the slanted edge method.

3 SLANTED EDGE METHOD

3.1 Motivation

The approach is based on the idea that it is difficult to obtain the Point Spread Function (PSF) of a camera/lens taking a capture of a single distant star, a POINT, against a black sky because of the relative intensity and size of the target [1].

Simply put, PSF is the impulse response of the optical system, i.e., if we were to capture the image of that star, then the image of the star as represented in our captured image will be a discretely sampled version of the PSF [1]. Because the imaging system is assumed to be linear, one can build intensity up and reduce noise by capturing a number of closely spaced stars in a row (a straight LINE) to obtain the MTF in the direction perpendicular to the line. Even better would be capturing a number of contiguous lines of stars, which at this point can be considered as a white EDGE against a darker sky [1] (Fig.1).

Fig. 1. A Slanted Edge

Another aspect of this concept would be in terms of spatial frequency. Nyquist-Shannon sampling theory implies that the input signal can be represented exactly using a series of samples spaced 1/(2F) seconds apart. This can be translated in terms of spatial frequencies as cycles per pixel or cycles per millimetre.

Next, an image captured by a digital image sensor can be sketched as a grid of dimensionless point-sampled values. The fact that the actual sensor pixels are not points, but the actively sensing part of each photosite (its aperture) does approximately look like a little square. This can be regarded as lumping the effect of this non-point like pixel aperture with the spatial effects of the lens. The image so obtained is a sampled system PSF which is the convolution of the photosite aperture PSF and the lens PSF. This makes it clear that the sampling process is a "pure" point-sampling process, exactly like the way it is interpreted in the Nyquist-Shannon theory.

For the correctness of the theorem, the requirement, that the signal power is maintained zero at frequencies above 0.5 cycles per pixel, has to be met. But interestingly enough, the Nyquist-Shannon sampling theorem does not tell what will happen if the input signal does not meet the requirement. In such cases, the effect of Aliasing can be witnessed in the frequency domain: energy at frequencies above Nyquist will be "folded back" onto the frequencies below Nyquist. But aliasing does not destroy information or detail; it merely makes it hard to tell whether the energy of the signal is at frequency f.

And this leads to the crux of the matter: even though the image sensor is sampling at a rate of one sample per pixel, it does not mean that the information has been lost irretrievably. The next task would be to identify which frequencies are aliases, so as to effectively undo the aliasing. One such scheme is to assume the structure of some part of the image, for example, an ideal knife-edge target, slanted at some angle. With the assumption of this prior image structure, the undersampled image can be unscrambled to recover the contrast at frequencies well beyond the Nyquist limit. A perfectly straight edge of infinite contrast is chosen for this method to work accurately.

3.2 Edge to Line to Modulation Transfer Function

The slanted-edge algorithm automatically computes the MTF from a user-defined rectangular ROI which represents a near-vertically or near-horizontally edge. The algorithm described as per norm ISO 12233:2017 assumes a near-vertical edge.

To measure near-horizontally, the selected edge image data is rotated 90° before performing the calculation. In general, the terms MTF and the Egde Spatial Frequency Response (e-SFR) are used interchangeably. The algorithm works as follows: In the start, the user selects the ROI containing the slightly slanted edge. The result is a two-dimensional matrix of data values (eg: n lines, m pixels). For each line of pixels perpendicular to the edge, the data is convoluted with a Finite Impulse Response (FIR) filter to get a one-dimensional derivative. After multiplying with the Hamming window vector for each data line, the edge location is estimated by finding the centroid to a subpixel accuracy. The result is a vector of centroid locations (l, n). A linear best-line fit to the centroid locations as a function of line number computes the initial estimate of slope and offset. The above process is repeated to obtain the best-fit slope and offset values. Using these computed values, the pixels in ROI are projected along the direction of the estimated edge to calculate perpendicular distances. These pixel values are sorted using bubble sort in the order of their distance from the edge. With the help of pixel-binning technique, this array of sorted pixel values are segregated into equal sized bins to reduce the influence of signal aliasing on the estimated MTF [2]. The values of the pixels collected in each bin are averaged, which generates a one-dimensional, over-sampled edge profile. In a nutshell, a two-dimensional edge is projected onto a one-dimensional line producing the edge intensity profile (ESF) in the direction perpendicular to the edge (Fig.2).

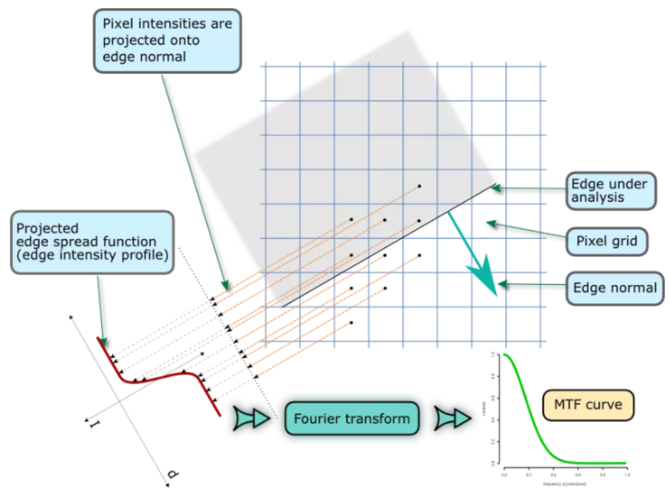

Fig. 2. Edge Spread Function

Second, a one-dimension derivative of this ESF is calculated to attain the Line Spread Function (LSF), which is directly proportional to the intensity profile of the system's 2D PSF in the direction perpendicular to the edge/line (Fig.3).

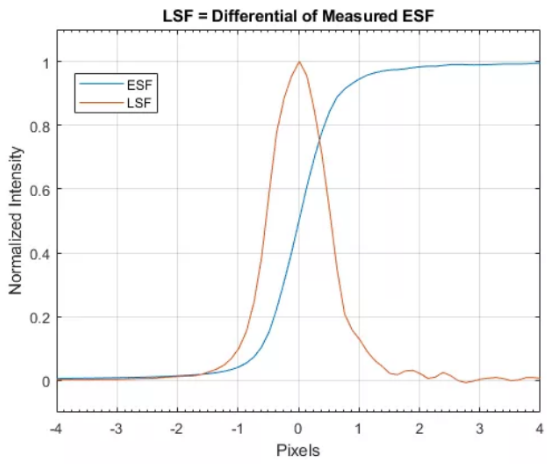

Fig. 3. Line Spread Function

Third, after applying a discrete Fourier Transformation to LSF, the magnitude (modulus) of the result so obtained is normalized to determine MTF. (Fig.4).

Fig. 4. Modulation Transfer Function

Slanted edge method enjoys several concrete advantages over the others. It effectively super-samples the edge by the number of pixels along it. This allows to take a much closer look at an approximation of the continuous ESF profile of the system on the imaging plane than possible with visual charts. Although, this method exhibits advantages of target simplicity and a small test image area, it demonstrates pitfalls of alignment sensitivity. In a digital sampling system, it is difficult to get normal edge profiles for non-sampling-aligned edges without the need to interpolate or otherwise introduce sampling errors. Therefore, most algorithms do not attempt this, assuming that the edge is near aligned to the sampling grid. Further, the ISO 12233 method is not robust against random noise intrusions. Errors in the estimation introduce a negative bias into the computed MTF, resulting in an underestimation of the actual MTF.

4 CPU vs GPU

Both CPU and GPU are both silicon-based microprocessors and have a lot in common. But at the same time, they are substantially different and are deployed for different roles. While CPU consists of a few cores that are optimized for sequential signal processing and maximizing the performance of a single task within a job, a GPU uses thousands of smaller and more efficient cores for a massively parallel architecture aimed at handling multiple functions at the same time.

As GPU is designed to execute the same operations on each item of work such as a pixel of an image, or an element of an array, it can be conceptualized as a large Single Instruction Multiple Data (SIMD) processor array supporting data-parallel applications. Applications most suited to this programming model are ones where there is little dependency between the data items in the computation, such as vector multiplication for machine learning algorithms or rendering graphics to a screen. By virtue of its parallelism, a GPU implementation is much faster than CPU implementation to estimate MTF.

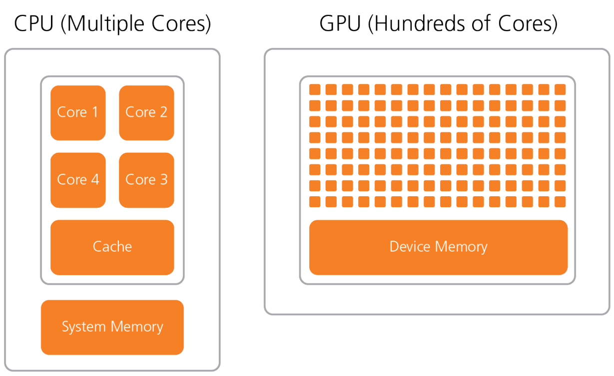

Fig. 5. CPU vs GPU

5 IMPLEMENTATION

5.1 Implementation on CPU

Pre-processing of image data in CPU is implemented in a two step process: Region extraction and Edge detection. In context to ISO standard 12233:2017, 2

CT images named h_edge.tif and v_edge.tif are operated upon to determine a user-defined ROI containing the slanted horizontal edge and vertical edge respectively. The output generated is a 2D matrix of data with rows corresponding to number of lines and columns to the number of pixels.

After the region extraction, an initial estimation of the location of edge parameters - slope and offset is predicted. Minimizing edge detection errors like noise suppression is most critical for getting sensible results. One way of doing this is to attenuate some of the artifacts resulting from the application to a signal of a finite length. A common windowing function called Hamming window of size N is used for this purpose, which can be characterized by equation 1:

$$V = 0.54 + (0.46 * cos\frac{2\pi n}{N}) \;\; where\; 0 \leq n \leq N \tag{1}$$

For a 2D matrix image, each line of pixels is convoluted with a Hamming filter [-0.5,0.5] to compute a one-dimension derivative. The resultant array is multiplied with the hamming window vector generated above to smoothen the values. The output of convolution and multiplication with Hamming window (Refer Fig: 6)is displayed in Section 6.To compute the location of edge for each line of pixels, a one-dimensional centroid of this matrix is calculated line by line. The centroid equation for each row is given by 2:

$$Centroid = \sum_{i=1}^{m} \frac{i \;\; * \;\; pixel\; intensity}{pixel\; intensity} \tag{2}$$

Finally, the slope and offset values are estimated using a linear fit function which calculates the array of centroid locations as a function of line number using the line equation $y = a_0x^0 + a_1x^1$ [2].

The final estimation of edge location is done using the back substitution method. The initial estimate values of slope and offset are substituted for each line number to obtain Hamming window array. The derivative of this array after convolution is used to generate a one-dimension vector of centroid locations, which is then passed to the linear fitting function to obtain the final slope and offset [2].

5.2 Implementation on GPU

In image processing, faster computation on pixel values is needed. As GPU is able to process pixels in a parallel fashion, faster execution times can be realized in GPU compared to CPU implementation. A one-dimensional super-sampled ESF is formed using the data of the truncated two-dimensional ROI image data. With reference to the previously calculated values of slope and offset, the pixels in ROI are projected along the direction of estimated edge by the equation 3

$$Distance = \frac{A * i + B * j + C}{\sqrt{(A)^2 + (B)^2}} \tag{3}$$

where A, B and C are slope, constant value of -1 and offset respectively. i and j represent the pixel positions.

To reduce complexity, Sorting and Data binning of pixel values is implemented in CPU. In pixel-binning technique, with the pixel values as reference points, the pixels are grouped in equal sized bins. This process is particularly useful to smoothen the ESF curve. The pixel values of each bin are then averaged and the resultant pixels are plotted to obtain the super-sampled ESF.

The ESF is differentiated to obtain LSF. Fast Fourier transform (FFT) of the generated LSF results in a complex array. The magnitude of this array is normalized to produce MTF curve [2]. A python script is then used to plot the three curves.

6 RESULTS AND CONCLUSION

6.1 Output

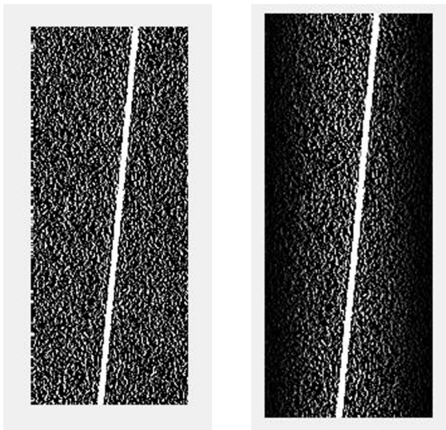

Fig. 6. Convolution of Image data and Hamming Filter, Multiplication of Convoluted Image with Hamming Window Vector

ESF of the ROI containing the slanted horizontal and the vertical edge is plotted as shown in Fig.7 and Fig.8. The plot for both PSF and MTF remains the same as depicted in Fig.9 and Fig.10 respectively.

6.2 Performance Evaluation

The performance of the system is evaluated using 2 metrics, execution time and accuracy. Table 1 and Table 2 represents the execution time of the hybrid model to implement MTF.

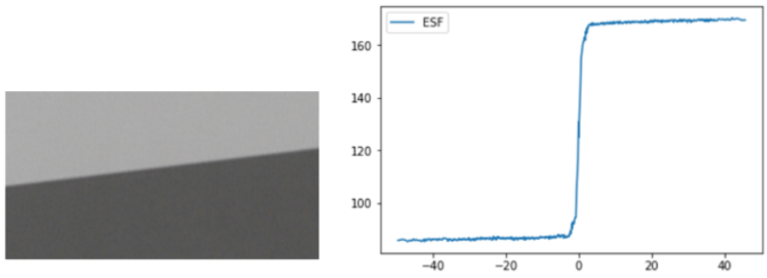

Fig. 7. Edge Spread Function for slanted horizontal edge

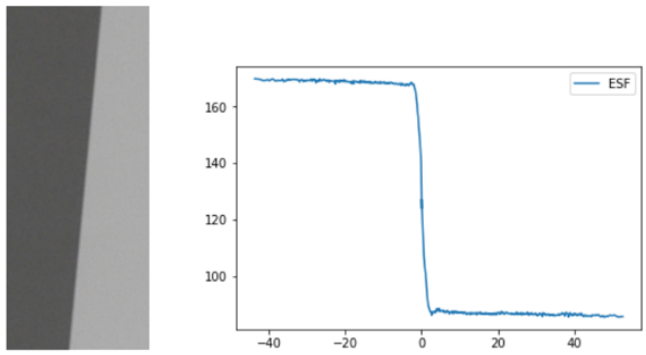

Fig. 8. Edge Spread Function for slanted vertical edge

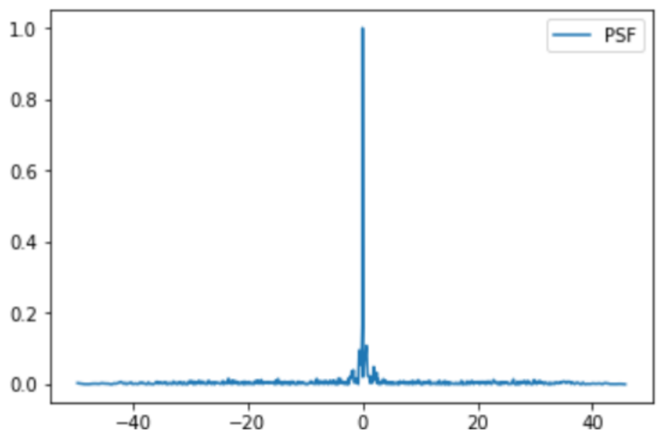

Fig. 9. Line Spread Function

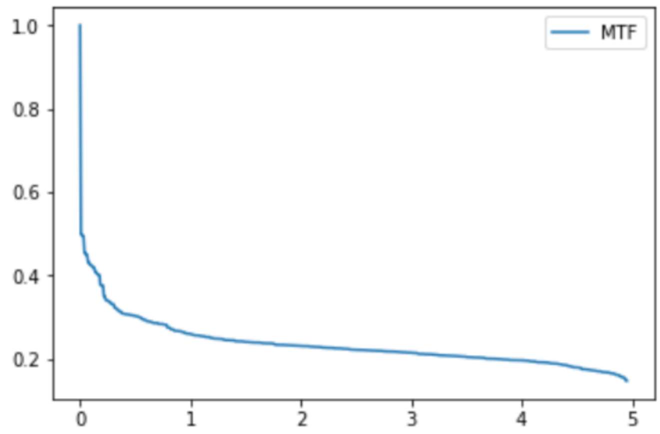

Fig. 10. Modulation Transfer Function

		CPU Time (in sec)	GPU Time (in sec)
ESF	Pre-processing	0.010669	N.a
	Edge-distance Calculation	N.a	0.000051
	Bubble Sort	14.614	N.a
	Data binning	0.000389	N.a
PSF	Derivative Calculation	N.a	0.000005
MTF	FFT Calculation	N.a	0.000003
	Magnitude	0.000030	N.a
	Bubble Sort	0.018531	N.a
	Total time	14.644	0.000059

Table 1. Hybrid (CPU + GPU) Execution time using Bubble sort technique

		CPU Time (in sec)	GPU Time (in sec)
ESF	Pre-processing	0.010669	N.a
	Edge-distance Calculation	N.a	0.000051
	Quick Sort	0.010287	N.a
	Data binning	0.000389	N.a
PSF	Derivative Calculation	N.a	0.000005
MTF	FFT Calculation	N.a	0.000003
	Magnitude	0.000030	N.a
	Quick Sort	0.000259	N.a
	Total time	0.021634	0.000059

Table 2. Hybrid (CPU + GPU) Execution time using Quick sort technique

The hybrid execution of MTF calculation executes by switching the computation between CPU and GPU. From the results, it is clear that GPU execution takes a constant time of 0.000059 seconds. The total execution time with the

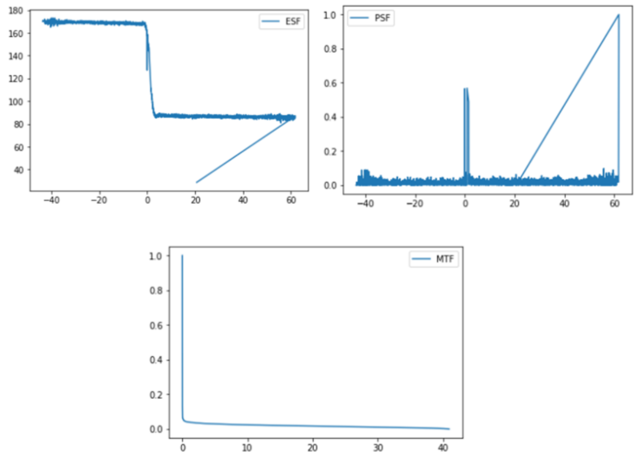

Fig. 11. Calulation of ESF, LSF and MTF using unequal sized bins

hybrid implementation using bubble sort takes around 14.644 seconds [Table1]. However, using a faster sorting algorithm i.e quick sort reduces the total execution time to 0.021693 seconds [Table2]. Thus, implementing a faster sorting algorithm reduces the total execution time by approx 635 times. The MTF calculations with the MATLAB code takes an average time of 0.056897 seconds.

For accuracy, Root Mean Squared Error (RMSE) is used as a metric. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. A lower values of RMSE indicates better fit and higher accuracy. The RMSE measured for the MTF calculation using the hybrid technique and Matlab, deviates by a value of 0.1162. Even though the hybrid execution drifts from the Matlab results by 11%, it remains to execute faster by 1.62 times.

Implementation of MTF calculation using data-binning, described in ISO standard 12233:2017 assumes equal sized bins. However, when this approach is implemented with unequal sized bins, the algorithm calculations are disproportionate. This can be observed from the following Fig.11.

References

[1] Jack: The slanted edge method (2014) (`https://www.strollswithmydog.com/the-slanted-edge-method/`)

[2] 42, I.: Iso 12233:2017 (2017)