A Software Tool for Measurement of the Modulation Transfer Function

Yen-Hong Kao[a], Michael Albert[a], Ann-Katherine Carton[a],
Hilde Bosmans[b], Andrew D. A. Maidment[a]
[a]Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104
[b]Katholieke Universiteit Leuven, Leuven, Belgium

ABSTRACT

Calculation of the modulation transfer function (MTF) is a multi-step procedure. At each step in the calculation, the algorithms can have intrinsic errors which are independent of the imaging system or physics. We designed a software tool with a graphical user interface to facilitate calculation of MTF and the analysis of accuracy in those calculations. To minimize the source of errors, simulated edge images without any noise or artifacts were used. We first examined the accuracy of a commonly used edge-slope estimation algorithm; namely line-by-line differentiation followed by a linear regression fit. The influence of edge length and edge phase on the linear regression algorithm is demonstrated. Furthermore, the relationship of edge-slope estimation error and MTF error are illustrated. We compared the performance of two kernels, [-1,1] and [-1,0,1], in the computation of the line spread function (LSF) from finite element differentiation of the edge spread function (ESF). We found that there is no practical advantage in choosing the [-1,0,1] kernel, as recommended by IEC. However, a correction for finite element differentiation should be applied; otherwise, there is a measurable error in the MTF. Finally, we added noise into the edge images and compared the performance of two noise reduction methods on the ESF; convolution with a boxcar kernel and a monotonicity constraint. The former method always produces MTF error higher than 4% up to the sampling frequency, while the latter was consistently less than 1%.

Keywords: Modulation transfer function (MTF), edge spread function (ESF), line spread function (LSF), edge slope estimation.

1. INTRODUCTION

The modulation transfer function (MTF) describes the spatial resolution characteristics of an imaging device, and is essential for quantifying the performance of the device. The MTF measurement starts with imaging a straight edge placed at a small angle (1.5^{o} ~3^{o})[1] to the pixel matrix array. From this digital image, the exact angle of the edge is detected and the distance of individual pixels to the edge is computed to construct a super-sampled edge spread function (ESF). Differentiation of the ESF generates a line spread function (LSF), whose Fourier transform produces the MTF[2-5].

Each step in the MTF calculation involves numerical computation, and can have intrinsic errors. In addition, factors such as edge length (image size) and edge angle can affect MTF accuracy. These errors are inherent in the MTF algorithm, and are not related to noise of the imaging system, nor the scattering due to the imaging physics[6], nor any artifacts. Initially, simulated ideal edge images without any noise were used to test the performance limits of edge detection, ESF construction, LSF numerical differentiation and discrete Fourier transformation. Later, realistic system noise was added to compare robustness of two ESF conditioning algorithms; convolution with a boxcar kernel and a monotonicity constraint.

Accuracy of edge angle detection is critical for MTF calculation. We tested the accuracy of a common detection algorithm: differentiate individual columns or rows, depending on which is nearly perpendicular to the edge; find the local maximum per line; and apply a linear regression fit to obtain the angle. This algorithm was applied to edge lengths ranging from 14 to 128 pixels, and on various simulated edge angles from 1.5^{o} to 4^{o}. More importantly, we demonstrated how edge angle detection error could affect the MTF accuracy. We also compared two types of kernels, [-1,1] and [-1,0,1] for differentiation of the oversampled ESF to produce LSF[1]. The effect of correction for this

Medical Imaging 2005: Physics of Medical Imaging, edited by Michael J. Flynn,
Proceedings of SPIE Vol. 5745 (SPIE, Bellingham, WA, 2005)
1605-7422/05/$15 · doi: 10.1117/12.597323

numerical differentiation was also demonstrated[7]. Finally, we applied our MTF algorithm on a perfect edge image, removing all possible algorithm errors, to illustrate the optimal numerical MTF attainable.

To expedite analysis of our MTF algorithm accuracy, we designed a software tool with a graphical user interface in Matlab (Mathworks, Version 7 R14, Natick, MA). After rigorous testing and validation of our MTF algorithm, we intend to distribute it freely with the hope to facilitate accurate MTF analysis and routine MTF measurements of medical imaging systems. The analysis presented here is part of a larger study in which we compare the performance of various edge-slope estimation algorithms, ESF binning methods, and noise reduction methods.

2. EDGE IMAGE SIMULATION

Edge images were simulated using a method similar to that presented by Carton et al.[8, 9] Two types of images were constructed, one without any type of noise or scattering, the other with realistic system noise. Both types assumed a linear response of the detector. Thus, the signal intensity in each pixel is proportional to the area covered by the edge device. If each pixel is square and has dimension $\omega \times \omega$ (ω=100μm), then as the edge travels through each pixel at an angle θ, there is a constant shift of $\omega \cdot \tan(\theta)$ in the edge position. However, the edge coverage area (ECA) in adjacent pixels does not always change linearly. For example, in figure 1, the edge starts at the exact lower left corner of the pixel in column 1, which has a triangular ECA. As the edge enters columns 2 and 3, the respective ECAs increase by the multiple of the area of the same parallelogram. In column 4, the edge shift of $\omega \cdot \tan(\theta)$ crosses into the neighboring pixel above. The ECA of the lower pixel has increased by the area of a polygon (figure 1), while the upper pixel has an ECA that is triangular. The calculation used in this paper for each condition is addressed in detail in the work of Carton et al.[8, 9]

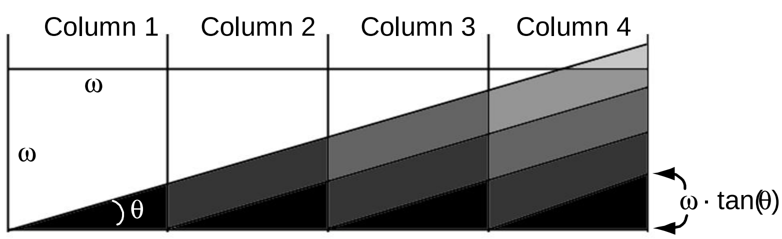

Figure 1. Schematic diagram of edge coverage area in adjacent pixels. Shades of gray represent differences in the coverage area by the edge device.

3. EDGE-SLOPE ESTIMATION

Our simulated edge travels horizontally along the row direction. In this orientation, differentiation along each column with a [-1,1] kernel is performed. The point with maximum value after differentiation is taken as the position of the edge for the column. A least-squares linear regression fit to a first-degree polynomial $Y=aX+b$ is performed using all detected edge positions. The coefficient a is the slope of the edge and is equivalent to $\tan(\theta)$. The number of columns the edge travels before it shifts into the next row, n, is $1/\tan(\theta)$ rounded to the nearest integer.

We tested the accuracy of this edge-slope estimation algorithm by using edge lengths of 14 to 128 pixels at 1.5°, 2°, 2.5°, 3°, 3.5°, and 4° (figure 2). The simulated edge for all images started precisely at the lower left corner of the first edge pixel, as in figure 1. Because there is no noise in the images, edge detection was always correct. However, the precision of this edge detection algorithm, namely the line-by-line differentiation, is at best one pixel (an integer), while the actual edge position is subpixel (a fractional number). For example, in figure 1, the edge positions from column 1 to column 3 all fall in the same row, but the actual position of the edge for each column differs by $\omega \cdot \tan(\theta)$. Subpixel precision of the edge position is achieved by the linear regression fit, which requires a certain number of data points to generate an accurate fit. Considering the situation in which the edge length equals n, in a certain edge phase arrangement (see discussion below), the line-by-line differentiation edge detection algorithm would generate edge positions on the same row. A linear regression fit of these edge positions would result in a slope of zero. Therefore, edge positions obtained by the linear regression fit of edge lengths close to n are always erroneous. The accuracy improves as the ratio of the edge length l to n, increases (figure 2). This algorithm favors a larger edge angle θ, because n is inversely related to θ. For instance, when the edge length is 75 pixels, the edge-slope estimation error on a 4^o edge is less than 0.1^o (l/n = 5.36), but on a 1.5^o edge, the error is still about 0.3^o (l/n = 2.86).

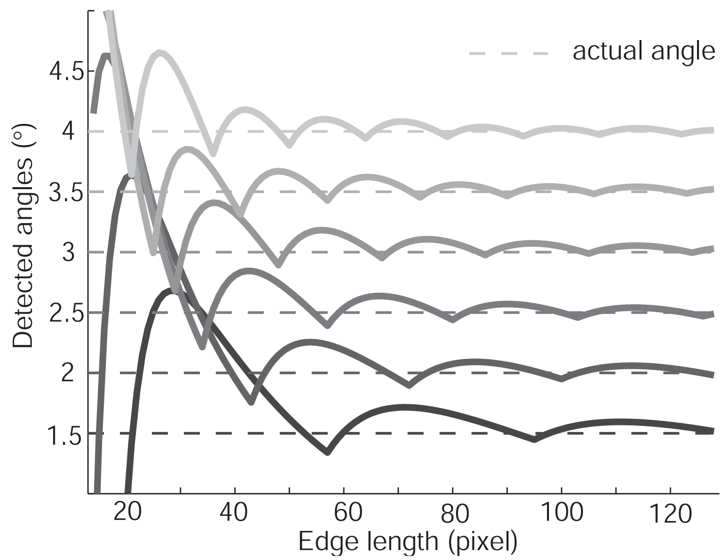

Figure 2. Effect of edge length on accuracy of edge-slope estimation analysis. Smaller edge angles require longer edge lengths for accurate estimation.

An edge does not necessarily start in the same phase each time; there are infinite ways to position the edge in the very first pixel. We investigated how edge phase affects the edge-slope estimation algorithm. We labeled each edge phase according to the percentage of ECA in the first edge pixel. Four phases were tested, 12.5%, 37.5%, 50% and 87.5%. Figure 3A shows that this slope detection algorithm is highly affected by starting edge phase. For instance, at the phase of 50% ECA, this algorithm constantly underestimates the edge slope. At 87.5% phase, the detected slope changes in a cyclic pattern nearly centered about the actual slope. The underlying cause for the phase effect is illustrated in figure 3B. Using an edge that spans exactly 6 pixels and starts at precisely the lower left corner, the detected edge positions are distributed over two rows. At 50% ECA, the detected edge positions are all on the same row. The combination of starting edge phase and the edge length contribute to the cyclic pattern of the detected edge slope. However, as the edge length increases to $l/n > 5$, the detected slope converges to within 0.1^o of the actual slope, although it remains cyclic.

In order to achieve accurate edge-slope estimation using the algorithm of line-by-line differentiation followed by a linear regression fit, an edge length having l/n ratio larger than 5 is recommended. For an edge slope of 3^o, this translates into an edge length of 96 pixels, and 1.5^o, 191 pixels.

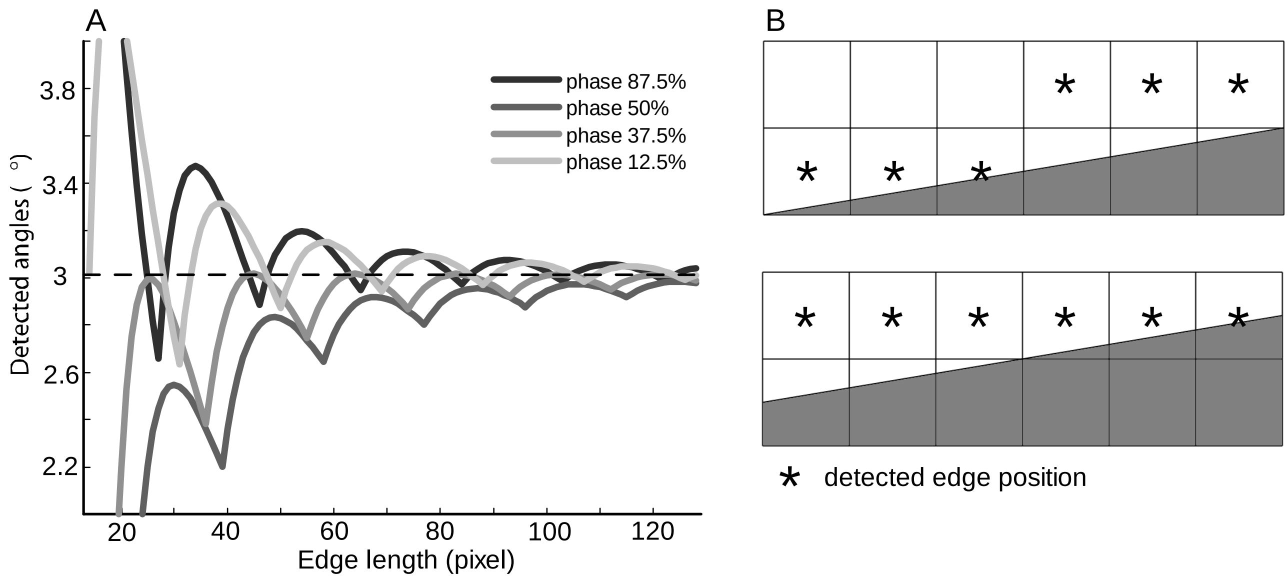

Figure 3A. Edge phase affects the accuracy of the linear regression edge slope detection when l/n <5. Longer edge lengths should be used to reduce the edge phase effect. **B.** Edge position detection is affected by edge phase. Top, an edge that spans exactly 6 pixels in one row results in two rows of detected edge positions. Bottom, the same edge shifted by half a pixel height yields detected edge positions all located in the same row.

4. ESF, LSF AND MTF

Once the slope of the edge is known, the precise edge position for each column can be calculated. The ESF are calculated as follows: the vertical distance between the center of each pixel and the edge is computed and rounded to the nearest integer; the signal intensities of the pixels of the same distance from the edge are binned together; the number of pixels per distance bin is recorded; the binned intensity is then divided by the total number of pixels of the corresponding distance bin. To generate a more finely sampled ESF, the calculated distance of each pixel from the edge is multiplied by a super-sampling factor, s, an integer number up to n. This operation is equivalent to sampling the edge with pixel spacing of (ω/s). For the analysis in this paper, a super-sampling factor of $s=n$ is used. A super-sampling factor of 10 has been used by others for edge images with noise[3].

The LSF is obtained by convolving the ESF with a [-1,1] kernel. The resulting LSF is then truncated at both ends to an integer number that is a power of 2. A discrete fast Fourier transform[10] is applied to the truncated LSF to obtain the MTF. A correction for finite element differentiation is then applied to the MTF[7]. Another option for computing LSF is the convolution with a [-1,0,1] kernel, as recommended by IEC[1]. We compared the effect on the MTF from these two kernels and found the difference to be minimal; on the order of 10^{-14}, as expected analytically. However, the correction for the [-1,0,1] kernel has a singularity, resulting in division by zero at ½ the super-sampling frequency. In either case, the correction for finite element differentiation should be applied; otherwise, the resulting MTF would be overestimated, especially beyond 3 times the sampling frequency (figure 4). The error in the area under the MTF without finite element differentiation correction up to the sampling frequency is 0.04%.

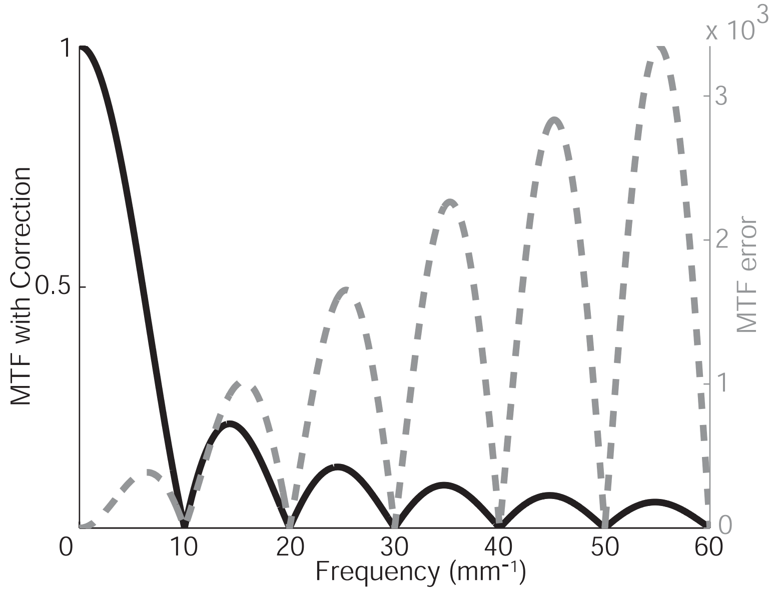

Figure 4. The MTF without correction for finite element differentiation results in measurable errors (dashed line). Corrected MTF (solid line). Edge slope = 2.05°.

5. COMPARISON OF ANALYTICAL MTF AND CALCULATED MTF

When a rectangular aperture is used to sample a line at angle $\theta = 0^{o}$, the resulting LSF is a rect function (figure 5, left). The Fourier transform of the rect function is a sinc function. When the sampling aperture is at an angle to the line, the resulting LSF has the shape of an isosceles trapezoid, which is equivalent to the convolution of two rect functions (figure 5, right). Its corresponding Fourier transform is the product of two sinc functions. In our analysis, the analytical MTF for a perfect edge placed at angle θ, and imaged with a noiseless linear system is

$$\mathrm{MTF}(f) = \left|\operatorname{sinc}(\omega \cdot f \cdot \sin(\theta)) \cdot \operatorname{sinc}(\omega \cdot f \cdot \cos(\theta))\right|,$$

where ω is sampling aperture size and f is the frequency.

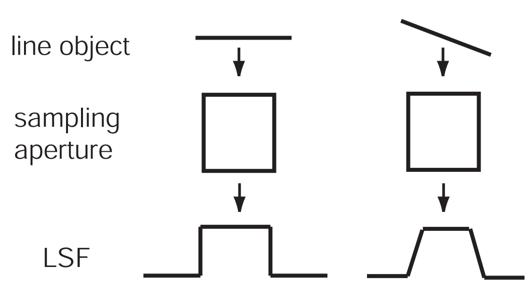

Figure 5. Schematic diagram of LSF from line objects sampled perpendicularly to the aperture (left), or at a small angle (right).

To assess the performance of our MTF algorithms, we computed the difference between the calculated MTF and its corresponding analytical MTF. For example, we considered a noiseless edge image in which the edge was placed at an angle that took precisely 28 pixels to complete one phase (n = 28). This angle was chosen to reduce edge phase effects during computation of distance bins for the ESF. The entire edge length was 112 pixels (l/n = 4). The edge angle was hard-coded into the program in order to avoid MTF errors due to edge-slope estimation. Under these ideal conditions, the difference between the calculated MTF and the analytical MTF is on the order of 10^{-4}. The computed MTF consistently underestimated the analytical MTF. The cumulative error expressed as a percentage of the total area is 0.06% up to the sampling frequency (figure 6) and 0.01% up to ½ the sampling frequency.

We performed the same comparison of analytical vs. calculated MTF, without edge-slope estimation error, on three different edge slopes, 1/28, 1/19 or 1/14, which respectively corresponds to edge angles of 2.05°, 3.01° and 4.09°, with varying lengths from 14 to 128 pixels (figure 7). The edges all start with the same phase, from exactly the lower left corner of the first edge pixel. For each individual slope, the MTF error stays constant for all edge lengths above n. However, the MTF error is less for smaller edge slopes. For example, the MTF error for an edge slope of 1/28 (2.05°) is 4 times smaller than that for an edge slope of 1/14 (4.09°). This result supports choosing smaller edge angles for MTF calculation, as the intrinsic MTF algorithm error is smaller.

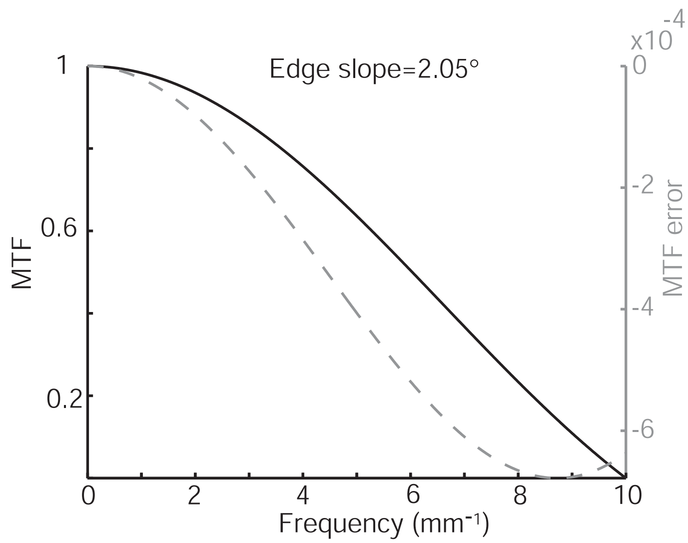

Figure 6. Analytical MTF (solid line). Difference of computed MTF from analytical MTF (dashed line).

One more comparison was performed to assess if different starting edge phases could affect MTF error. We calculated MTF errors on two edge phases, with respective starting ECA of 3.6% (line starts in the lower left corner) and 50%, for edge slope of 1/14 at varying edge length from 14 to 128 pixels (figure 7, “14 line” and “14 line starting ECA=50%”). The results indicate that when there is no edge-slope estimation error, the MTF calculation algorithm is phase invariant.

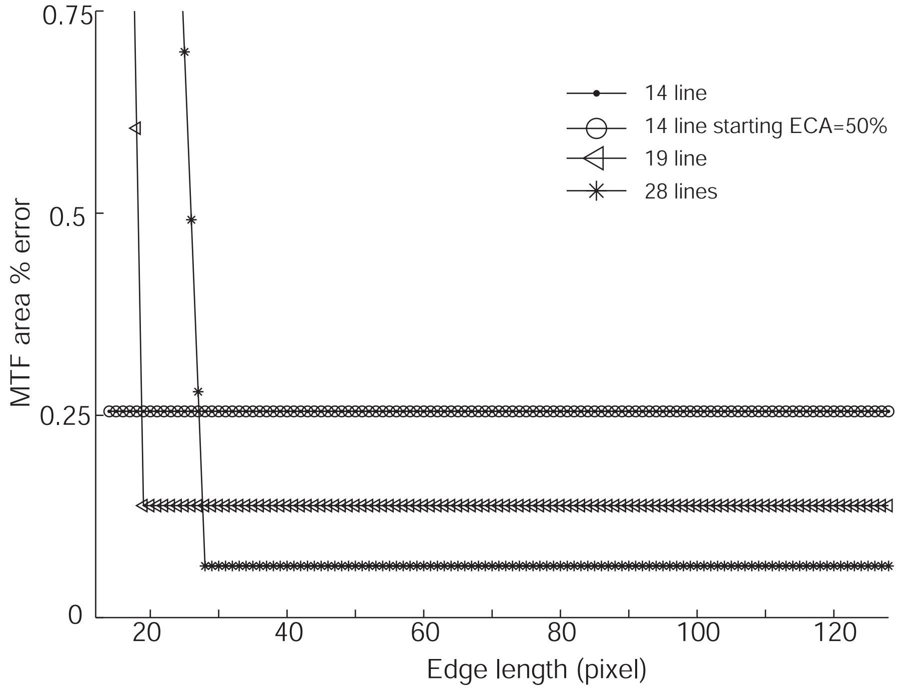

Figure 7. MTF error of three edge slopes. Smaller edge slope has less intrinsic MTF error.

6. EFFECT OF EDGE-SLOPE ESTIMATION ERROR ON MTF

An important question in the MTF calculation is how errors in edge-slope estimation affect MTF accuracy. We performed the analysis on three edges, each having a slope of 1/28, 1/19 or 1/14, which respectively corresponds to edge angles of 2.05^o, 3.01^o and 4.09^o. These choices ensure that the number of pixels for a complete phase of an edge (*n*) is an integer. We have chosen these values to avoid ambiguities in binning due to incommensurate lengths. This then allows us to investigate the edge phase effect on the MTF as the result of errors in edge-slope estimation. For this analysis, edge lengths of 14 to 512 pixels were used. Actual edge-slope estimation was not performed. The slopes were hard-coded and so were the corresponding slope errors at ±0.1^o, ±0.075^o, ±0.05^o, ±0.025^o and ±0.01^o. A gray-scale contour plot was made for each edge slope to demonstrate the relationship between edge-slope error and the resulting MTF errors at various edge lengths. The MTF error was computed up to the sampling frequency and expressed as a fraction of the total area.

In the contour plots (figure 8), each contour line represents an increment of 1% in error; for example, the white areas indicate the MTF errors from 0% to 1%, and next contour, from 1% to 2%. The completely dark area represents MTF errors larger than or equal to 10%. A distinctive feature of the contour plots is that there is a cyclic pattern in the relationship between MTF error and edge length, for all edge slopes and errors. In addition to the cyclic pattern, when the edge-slope estimation error is greater than 0.02^o, the MTF error increases more drastically as the edge length becomes longer. The increase in the MTF error is proportional to the edge-slope estimation error; the larger the slope error, the faster the increase in MTF error. For example, at an edge length of 350 pixels and an edge-slope estimation error of 0.1^o, the MTF error is around 10%, but when the length is increased to 512 pixels, the MTF error exceeds 20%. Conversely, for a slope detection error of 0.02^o, the MTF error for edge lengths of 350 pixels to 512 pixels only increases from just under 1% to slightly over 1%, for all three angles shown.

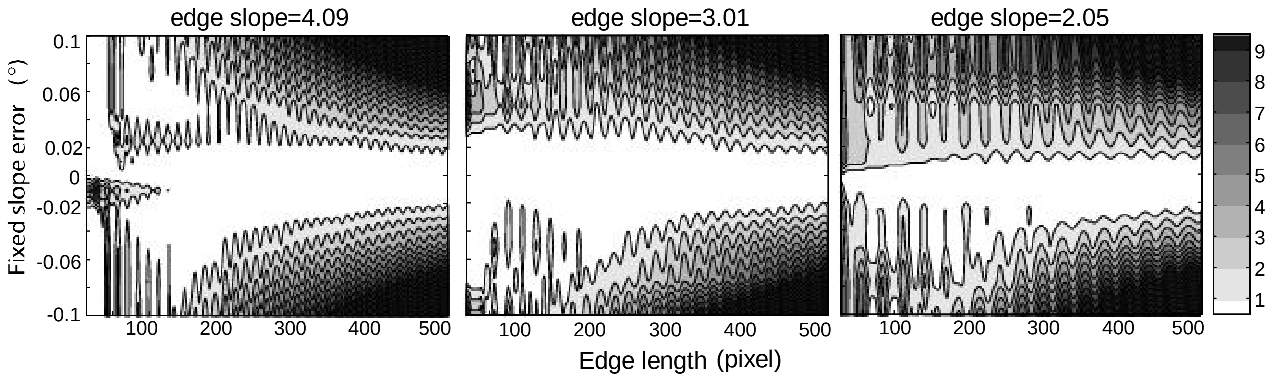

Figure 8. Combination effects of slope estimation error and edge length on MTF accuracy. White areas represent MTF errors < 1%, and black areas ≥ 10%. Larger edge detection errors, e.g. 0.1°, amplify MTF errors for larger edges. Edge detection errors of < ±0.02° and edge lengths > 150 pixels consistently produce MTF with errors < 1%.

These contour plots indicate that if the edge-slope estimation can be accurate to be within ±0.02°, when the edge length is greater than ~150 pixels, this algorithm can consistently produce MTF having less than 1% error up to the sampling frequency. However, when edge-slope estimation is prone to error, it is not recommended to use an edge length of longer than 250 pixels, as this MTF calculation algorithm could quickly amplify the slope error. The range of edge lengths of 150 to 250 pixels also corresponds to the widest vertical distance for the white areas in the contour plots while avoiding regions in which the MTF error is greater than 10%. The results of this analysis are not specific to the linear regression edge-slope algorithm, and can be generalized to all edge-slope estimation algorithms.

7. ESF DATA CONDITIONING

In real images, there is noise of both quantum and detector origin. We simulated the noise properties of the Embrace 1.0 CR system (Agfa, Mortsel, Belgium)[8] in our edge images. Without any signal processing, this noise greatly affects the accuracy of the MTF (figure 9, top, "none") especially in the higher frequency range. The resulting MTF curve has very little resemblance to the expected sinc function. The amplitude for frequencies beyond the sampling frequency can overshoot the amplitude at zero frequency.

We compared the effectiveness of two noise reduction methods; convolution with a boxcar kernel, and monotonic conditioning[11]. Both methods operate on the ESF. The boxcar method is the convolution of the ESF with a kernel of uniform amplitude. The size of the boxcar kernel in our analysis was equal to n, but can be set to any arbitrary number. Boxcar convolution removes a significant amount of noise. However, in the process of dampening the noise, it also artificially degrades the MTF (figure 9, top, "boxcar(19)"). The other method, monotonic conditioning, imposes a simple rule that the data in the ESF should be monotonic. It effectively removes noise without introducing a systematic error in the MTF; the shape of the sinc function is nicely preserved with monotonic conditioning (figure 9, top, "monotonic").

A comparison of the performance of the two noise reduction methods was made on noisy images with varying edge length. Again, the edge slope was hard-coded so as not to introduce confounding errors. The accuracy of the resulting MTF was measured up to ½ the sampling frequency. Monotonic conditioning consistently outperforms boxcar filtering; the accuracy of the former method is within 0.5% of the true MTF throughout the whole trial, while the latter method has error of up to 6%, and could never perform better than 4%. Monotonic conditioning is clearly the superior choice (figure 9, bottom).

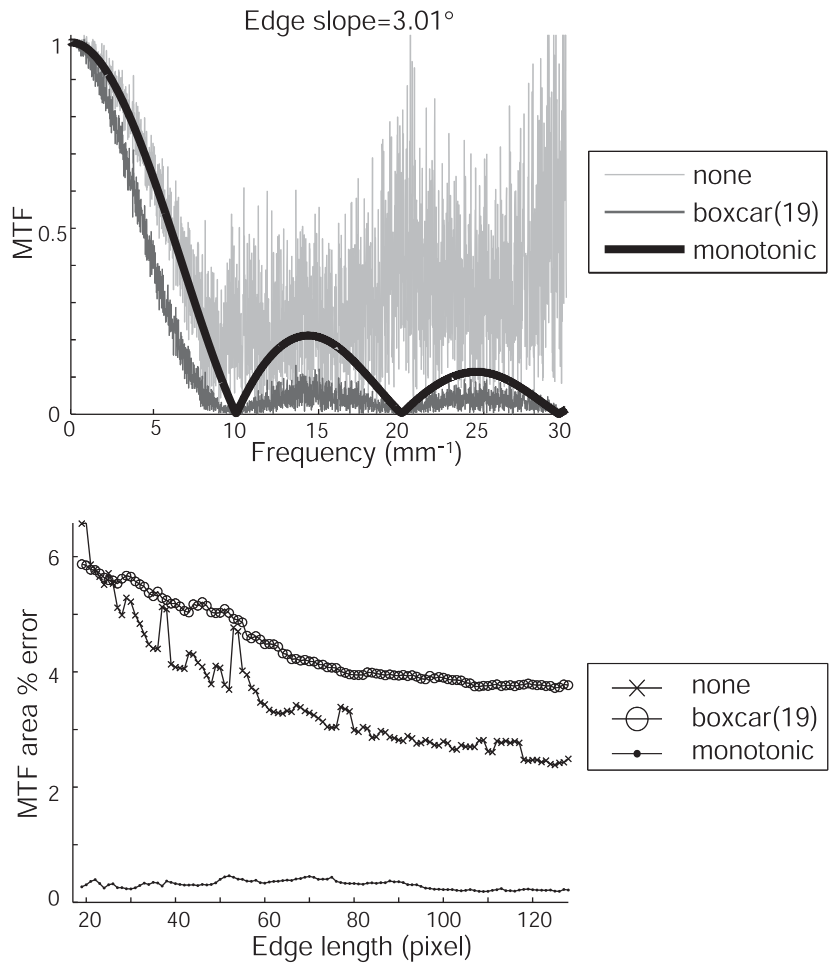

Figure 9. Comparison of MTF noise reduction methods. **Top**, monotonic constraint does not produce a systematic error. **Bottom**, monotonic constraint consistently produces MTF with < 1% error.

8. CONCLUSION

We have presented a detailed evaluation of methods for estimating MTFs. This is part of larger study comparing methods of quantifying detector performance. When the edge slope is estimated accurately, our MTF calculation algorithm achieves an accuracy of ~0.06% up to the sampling frequency, and 0.01% up to ½ the sampling frequency. Edge-slope estimation error is one of the main sources of MTF inaccuracy. When the edge length is > 250 pixels, a 0.1^o error in edge-slope estimation could result in > 10% error in MTF calculation. Edge-slope estimation errors of < 0.02^o with edge lengths > 150 pixels are required to consistently achieve a MTF accuracy of 1% or better. The optimal edge length for MTF calculation falls between 150 to 250 pixels. To minimize the edge-slope estimation error, edges having *l/n* ratio > 5 are recommended for the linear regression method. This recommendation also minimizes edge phase effects. Correction for finite element differentiation should be performed. However, there is no practical difference between the [-1,1] kernel and the [-1,0,1] kernel, although the [-1,0,1] kernel has a singularity at ½ the super-sampling frequency. Finally, monotonic conditioning of the ESF reduces noises in the data without introducing a significant systematic error in the MTF.

REFERENCES

1. "Characteristics of Digital X-Ray Imaging Devices-Part 1: Determination of the Detective Quantum Efficiency", *IEC 62220-1, Medical Electrical Equipment*, 2003.
2. P.F. Judy, "The line spread function and modulation transfer function of a computed tomographic scanner", *Medical Physics*, **3**(4):233-236, 1976.
3. E. Samei, et al., "A method for measuring the presampled MTF of digital radiographic systems using an edge test device", *Medical Physics*, **25**(1):102-13, 1998.
4. E. Buhr, et al., "Accuracy of a simple method for deriving the presampled modulation transfer function of a digital radiographic system from an edge image", *Medical Physics*, **30**(9):2323-2331, 2003.
5. H. Fujita, et al., "A Simple Method for Determining the Modulation transfer Function in Digital Radiography", *IEEE Transaction on Image Processing*, **11**(1):34-39, 1992.
6. P.B. Greer and T. van Doorn, "Evaluation of an algorithm for the assessment of the MTF using an edge method", *Medical Physics*, **27**(9):2048-2059, 2000.
7. I.A. Cunningham and A. Fenster, "A method for modulation transfer function determination from edge profiles with correction for finite-element differentiation", *Medical Physics*, **14**(4):533-537, 1987.
8. A.-K. Carton, et al., "Validation of MTF measurement for digital mammography quality control", *Medical Physics*, **In Press**, 2005.
9. A.-K. Carton, *Development and application of methods for the assessment of image quality and detector performance in digital mammography*,149, Katholieke Universiteit Leuven, Leuven, 2004.
10. M. Frigo and S.G. Johnson, "The Fastest Fourier Transform in the West", *MIT-LCS-TR*, **728**, 1997.
11. A.D.A. Maidment and M. Albert, "Conditioning data for calculation of the modulation transfer function", *Medical Physics*, **30**(2):248-53, 2003.