September 2008

MITRE Implementation of ISO 12233 Spatial Frequency Response (SFR)

Margaret A. Lepley mlepley@mitre.org
MITRE MTF and IQS Assessment mtf@mitre.org
The MITRE Corporation

The ISO International Standard 12233 [1] describes a method for computing the spatial frequency response (SFR[1]) of an imaging system from the image of an edge target that is slightly slanted with respect to the vertical or horizontal axis. Our goal was to implement code to perform this computation as specified in the standard.

This report includes a discussion of the SFR computation, implementation details for our code, and verification examples.

How should SFR be computed?

The SFR computation has 3 slightly different descriptions within ISO 12233[2]:

a. Section 6.3.2: text, flow chart, and figures
b. Annex A: *informative* source code in 'C'
c. Annex C: *normative* mathematical specification

Unfortunately these three do not always agree on the small details, and even Annex C is at times unclear and at times potentially contradictory. As further information we also downloaded the ISO 12233 Slant Edge Analysis Tool matlab code, sfrmat 2.0, from the International Imaging Industry Association (I3A) website [2], to help judge intent in places where the standard was less than clear.

This report documents the basic sections of a SFR measurement of this type, and lists how a, b, c, and sfrmat approach each of them. Finally the choice made for the MITRE_SFR code, which is trying to strictly adhere to the standard, is given.

The math behind all the concepts mentioned here is not provided, but a brief list of the key algorithm sections that are stable and those that vary is provided to enable the discussion that follows. The algorithmic segments that will be discussed are:

1. ROI: Region of Interest selection and error checking
2. OECF: Linearize image by use of the Opto-Electronic Conversion Function

---

[1] In this document, as in ISO 12233, the term SFR refers to the spatial frequency response obtained from the edge image output of an imaging system. It would be equivalent to the imaging system's Modulation Transfer Function (MTF) if the SFR computed from the image is divided by the SFR of the target edge, or if the target edge is a 'knife edge' with respect to the given imaging system, and if the imaging system does not exhibit any substantial frequency-independent contrast reduction (such as veiling glare). In other contexts the term SFR has a more general meaning, i.e., it is not tied to a specific target type.

[2] ISO 12233 is currently being revised to add a sine wave-based SFR measurement. The differences noted in the 12233:2000 edge-based SFR description may be corrected in this future edition. It is anticipated that the new version will not be approved and published until late 2007.

3. Centroid: Row centroids
4. Slope: Edge slope estimation
5. Integral number of phase rotations / Slope checking
6. Shift: Per row shifts
    a. Remove slope
    b. Shift to center
7. Bin: Oversample binning
    a. On Edge Spread Function (ESF) or Line Spread Function (LSF)
    b. Oversample factor
    c. Via rounding or truncation
    d. Handling gaps (zero bins)
8. Derivative: filter = [-1 1] or [-1 0 1]
9. Window width
10. Post-LSF centering
11. Hamming window
12. SFR: Normalized Modulus of Discrete Fourier Transform (DFT)
13. Finite Difference correction [3]

Of these, all the information is consistent in OECF, Hamming window, and final SFR computation. Annex A code includes unnecessary post-OECF normalization steps, but they do not impact the computation. For the other sections, the table below highlights differences, similarities, and points of ambiguity.

Table 1: SFR Description Comparison

	ISO 12233 Sect 6.3.2	ISO 12233 Annex A code	ISO 12233 **Annex C Normative**	Sfrmat 2.0
ROI Edge modulation	At least 20% on all edges	At least 0.2 on first & last edge	**No requirement**	At least 0.2 avg over all edges
Row centroids	Use [-1 0 1] derivative	Exact formula Using [-1 1] derivative	**Exact Formula Using [-1 1] derivative**	[-1 1] derivative, but includes other iterative operations + Hamming
Slope	Best fit line	Linear regression	Unclear formula	Linear regression
Shift per row (real number)	Use slope, reference to first row No avg centroid	Use slope, reference to center row No avg centroid	**Unclear:** Formula **references average centroid**, but missing '+' sign	Use slope, reference to first row No avg centroid
Rows used to compute binned LSF	✓	✓	**Integral num phase rotations within ROI**	All rows: Non-integral values allowed
Min number rotations	1 implicit	1	**1**	No check

Maximum slope(angle)	None explicit. In fig: 4 rows (|slope|=1/4)	Slope ≤1/4 angle < 14° rgt any left tilt OK	**None specified explicitly**	No check. Implicitly |Slope| < 1 (45°)
Binning	On LSF (incorrect) Scale=4 Gaps not mentioned	On ESF No rounding Scale=ALPHA Gaps = neighbor Bug in code	**On ESF With rounding Scale = 4 Gaps not mentioned**	On ESF No rounding Scale = 4 Gaps left
Derivative	[-1 0 1] in pixel domain according to text, while [-1 1] implied in Figure 5	[-1 0 1] in oversampled domain	Unclear: **[-1 1]** or [-1 0 0 0 1] in oversampled domain	[-1 0 1] in oversampled domain
Window width	✓	✓	**4X X=original width**	✓
Post LSF centering (integer bin)	None	Peak LSF	**None**	Centroid LSF
Finite Difference correction	None	None	**None**	None

There are four places where the Annex C normative reference is potentially unclear, as described in the following.

Slope formula. This formula isn't standard terminology for best-fit linear regression. But linear regression is believed to be the intent, and concurs with all software references.

Derivative filter. The scale of the indices, j, is not consistent throughout Annex C (sometimes j integral, sometimes j = n/4). However if equations C.8 and C.9 are read together, then it is clear that j should be treated as an integer, and the derivative filter is [-1 1].

Shifting. The value of the shift used impacts not only the binning results, but also whether post LSF-centering is required. For example, if an intercept offset is not included in the original shift, then some sort of centering is required prior to application of the Hamming window. Both the Annex A code and sfrmat use this technique of centering after the LSF has been computed. The Annex C description however does not make any mention of centering beyond the original row shift, which implies that the normative description of centering occurs during the shifting operation. This agrees somewhat with equation C.5, which includes an average centroid term as part of the shift. However, equation C.5 appears to be missing a necessary '+' sign. This makes it hard to state precisely what is intended at this point. Whether rounding is used during binning is a similar type of issue, i.e., rounding can be effected by altering the shift factors slightly, or vice versa. This is the only case where it is impossible to determine the precise intent of the standard.

Fortunately, all these variations in shifting and rounding, while they can cause large differences in the computed LSF, tend to have very little effect on the SFR within Nyquist range.

It should be noted that there are several areas to possibly improve this computation and make it more robust. Sfrmat2 includes some ideas in this direction, and the size of the final window might be another area of consideration. Such changes however would not match the normative description in Annex C.

MITRE_SFR Code

The computation within MITRE_SFR is derived from the code in ISO 12233 Annex A, but with some alterations to more closely follow the normative description in Annex C. MITRE_SFR follows the standard with respect to the ROI, OECF, integral number of cycles, Hamming window, and final SFR computation. Table 2 shows the basic functionality of MITRE_SFR relative to Annex A and C for the other functions.

The remainder of this section further details changes that were made from Annex A code which group into 4 basic categories: coding structure decisions, deleting unnecessary code, fixing bugs, and changing functionality to better match the normative description.

Coding decisions:

- Plain C pointers rather than handles
- Data array reading, rotation, and OECF correction in MITRE wrapper code

Table 2: Functionality of MITRE Code vs ISO 12233 SFR

	ISO 12233 Annex A code	**ISO 12233 Annex C Normative**	MITRE_SFR code
Modulation	≥ 20% first/last edge	**No requirement**	≥ 20% first/last edge (warning v1.3)
Row centroids	✓	**Exact Formula**	✓
Slope	Linear regression	Unclear formula	Linear regression
Shift per row (real number)	Use slope, reference to center row No avg centroid	**Unclear:** Formula **references average centroid**, but missing '+' sign	Use slope, center rows to avg centroid
Rows used to compute binned LSF	✓ First rows in ROI	**Integral num phase rotations within ROI**	✓ Centered in ROI (v1.3+ will run on less than 1 phase rotation: w warning)

Min phase rotations	1	**None specified**	5 (warning in v1.3)
Maximum slope	slope ≤1/4	**None specified explicitly**	Implicit slope < 1 |slope| ≤1/4 (≤v1.2) (info in v1.3)
Binning	On ESF No rounding Scale=ALPHA Gaps = neighbor Bug in code	**On ESF** **With rounding** **Scale = 4** **Gaps not mentioned**	On ESF No rounding Scale = 4 Gaps: As Annex A Bug fixed
Derivative	[-1 0 1]	Unclear: **[-1 1]** or [-1 0 0 0 1]	[-1 1]
Window width	✓	**4X** **X=original width**	✓
Post LSF centering	Peak LSF	**None**	None
Finite Difference correction	None	**None**	Yes [3]

Deleted code:

- Luminence extraction: only single band gray data is being used
- Normalization by the global MIN/MIX: Has no impact on SFR result
- All references to TestAbort

Bug fix:

- In the function bin_to_regular_xgrid, where the code fills zero bins by looking at previously computed bins. This code segment

```
if( (*counts)[i-k] !=0 ) {
     (*AveEdge)[i]=(*AveEdge)[i-k]/((double)(*counts)[i-k]);
     j=1;
}
```

  was changed to

```
if( (*counts)[i-k] !=0 ) {
     (*AveEdge)[i]=(*AveEdge)[i-k];
     j=1;
}
```

  since all earlier AveEdge bins have already been normalized by counts.
  This fix is low impact, since generally very few zero bins are found.

Changed functionality:

- Odd number of rows allowed in the ROI.
- Check to confirm that there are at least 5 full phase rotations (vs one).
- Rows eliminated from the region to get an integral number of phase rotations are evenly divided top and bottom, so that the center row location doesn't change.
- Row shifts were adjusted to include an average centroid offset, i.e.

```
offset = centroid[R/2] + 0.5 + best_fit_intercept – size_x/2;
```

```
(*shifts)[i] = slope * (i – R/2) + offset;
```

- Calculate_derivative was changed to use [-1 1] computation, specifically
```
(*AveEdge)[i] = (*AveEdge)[i] – (*AveEdge)[i-1];
```
- LSF Peak finding turned off. Instead,
```
pcnt2 = 4*(size_x/2);
```

Some other options were considered, but eventually not included:

- Derivatives: [-1 0 1] and [-1 0 0 0 1] with later finite difference correction of SFR
- Other row shift offsets: Offset = 0
- Bin_to_regular_xgrid using rounding vs truncation, i.e.
```
bin_number = (long)floor((*alpha)*(*edgex)[i] + 0.5);
```
  This is equivalent to adjusting row shifts by 0.125 and works best when used with a [-1 0 1] derivative computation.
- ESF window length: 4(X – num_cycles), where num_cycles = (int)(|slope|Y). This would ensure that full number of phase rotations contributed to all ESF bins, instead of having a decreasing contribution and hence greater noise at the bins furthest from the center.

Starting with code version 1.3, some checks that caused a halt in computations in earlier versions have been changed for greater flexibility:

- Minimum edge modulation. Less than 20% modulation causes a warning but processing attempts to continue. There is a high probability that there will be a failure later in processing due to the low contrast across the edge.
  Modulation = |leftmost grey – rightmost grey| / (leftmost grey + rightmost grey) is computed at the top / bottom of each ROI using 4-pixel average grey levels from the corners (for a vertical edge).
- Minimum/integral number of rotations. When angles are so acute that 5 phase rotations (or even one) are not present in the ROI, then a warning is printed, but processing continues.
- Maximum slope. Wide angles with an absolute slope bigger than ¼ are flagged, and when it is anticipated that this results in too little data for the oversampling, then a warning is printed, but processing continues.

Verification

The MITRE_SFR computer program was verified by processing digital simulation edges whose corresponding LSFs and SFRs are known to a high degree of accuracy, which establishes “ground truth” for each ESF-LSF-SFR test set. We independently developed a computer program, EdgeGen, to construct the digital simulation edges. In testing, simulated edge profiles whose SFRs approximately match the FBI’s single finger capture device specification [4] MTF were included, as were edges with undershoot/overshoot indicative of edge sharpening, nonlinear edges, and noisy edges, all at an assumed 500 ppi sampling rate, and upsampled edges. Having verified the MITRE_SFR code with these simulation edges, we then performed additional testing with real edges obtained from various imaging systems. Although the true SFRs of the real edges are not known, exercising the SFR code with real edges is insightful, as it gives a good sense of the proper operation of the code in the ‘real world’.

The basic steps in the EdgeGen program are as follows:

1) Compute one-dimensional (single row) edge profile from selection of available analytical functions (variables: edge shape, sharpness, and contrast)
2) Convolve the edge profile with a rectangle function representing a single pixel and sample the result every pixel width to get the sampled edge profile across 1 row (variable: pixel width)
3) Repeat the pixel convolution with the edge profile for all succeeding rows in 1 phase cycle, shifting the pixel location with respect to the edge by a subpixel amount for each succeeding row, which accounts for edge tilt with respect to the vertical (variables: tilt angle, pixel width). An edge tilt angle of 5.19443 degrees with respect to the vertical was used in all cases, which has a corresponding phase cycle of exactly 11 rows.
4) Optional: apply a nonlinear conversion to the single phase cycle edge profile in target space to obtain its equivalent profile in image space gray levels (variable: input/output equation representing the OECF)
5) Duplicate the single phase, multi-row edge profile from step 3 or 4 over multiple phase cycles, shifting 1 pixel along a row for each phase cycle (variable: number of phase cycles desired)
6) Optional: add gaussian noise to the edge image created in step 5 (variable: standard deviation of gray levels).
7) Write the constructed edge image to an output file in 8 bits/pixel PGM image format.

Concurrently with the above steps for edge profile generation, the program also computes the corresponding LSF and MTF, in most cases from analytical functions.

The true system MTF computed by EdgeGen is given by:

$$\frac{(\text{MTF}\ \text{corresponding to step 1 edge profile}) \times (\text{MTF}\ \text{corresponding to rectangle function})}{(\text{MTF}\ \text{corresponding to tilt angle})}$$

An upsampled edge is created by first processing through EdgeGen with a pixel width corresponding to a resolution less than 500 ppi, then using third party image manipulation software (Adobe Photoshop, NIH ImageJ, etc.) to rescale EdgeGen's output image to an equivalent 500 ppi scale. [This procedure produces an image edge with a scale of 500 ppi, but a resolution level less than 500 ppi.]

As an example, suppose the selected ESF is the error function, then the corresponding LSF and MTF are gaussians, i.e.,

$$\text{ESF} = \text{erf}(k_1x); \qquad \text{LSF} = \exp(-k_2x^2); \qquad \text{MTF} = \exp(-k_3f^2)$$

and the system MTF

$$\text{MTF}_{\text{system}} = \text{MTF}_{\text{edge}} \times \text{MTF}_{\text{pixel}} \,/\, \text{MTF}_{\text{tilt}}$$

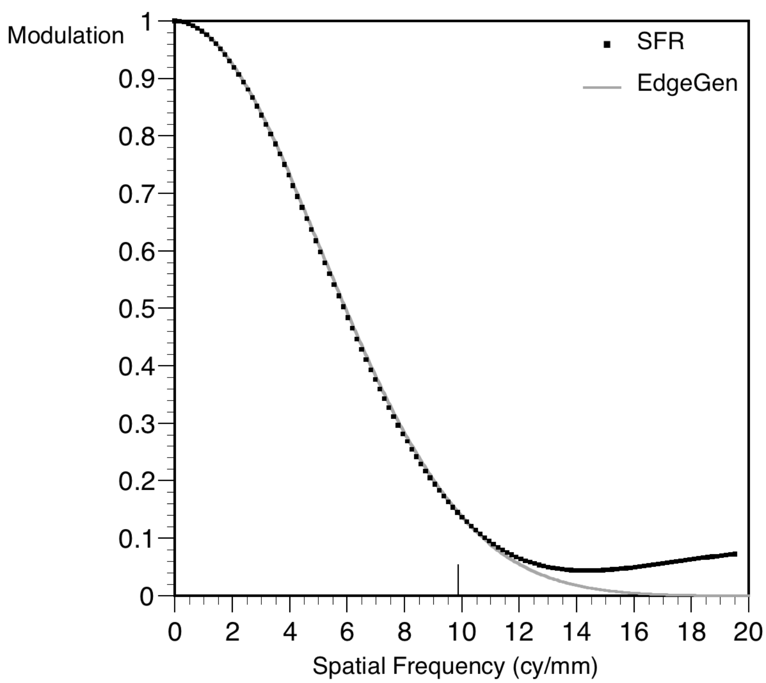
$$= \exp(-k_3 f^2) \ \text{x} \ \text{sinc}(\pi f w) \ / \ \text{sinc}(\pi f w t)$$

where: k=constant; x=distance; f=frequency; w=pixel width; t=tilt angle

The following plots compare the MTFs computed from SFR and EdgeGen for a selection of the verification cases run: no perturbations, sharpened edge, nonlinear edge, and noisy edge. Also included is a comparison between the SFR computed from a edge target and MTF computed from a sine wave target.

It should be noted that SFR or MTF values more than a few percent beyond the 9.84 cy/mm Nyquist frequency (500 ppi imaging system) are highly suspect on theoretical grounds and have arguable value; the MTF comparison from 0 frequency up to Nyquist frequency is our main interest for SFR code verification.

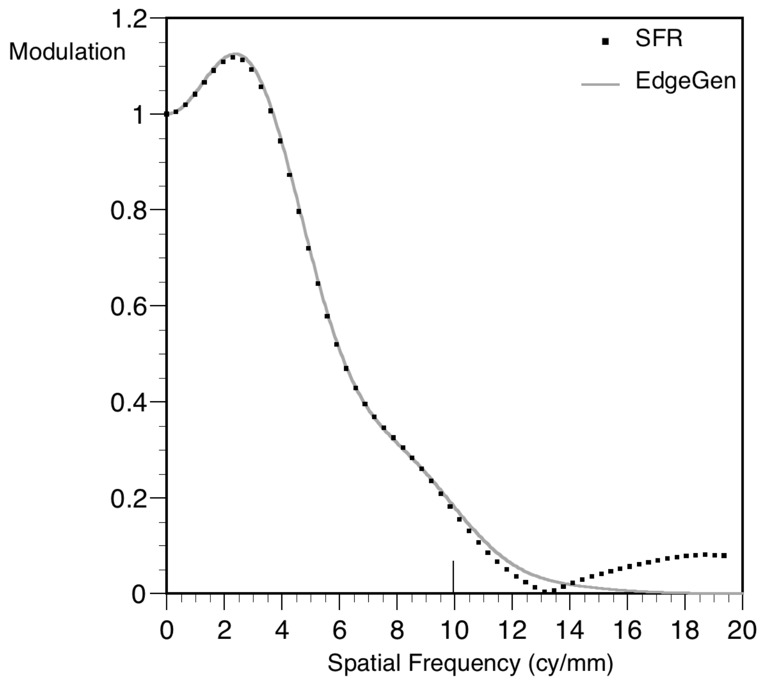
Case: No Perturbations



EdgeGen:
Max/Min Gray: 255/0
Noise: 0
Tilt Angle: -5.19443°
ppi: 500
pixel width: 0.0508 mm
Nyquist: 9.84 cy/mm
MTF type: Gaussian (fc=8.1)
OECF: linear

MITRE_SFR:
ROI width: 124
ROI height: 200
Calculated Tilt Angle: -5.194°
# of Phase Cycles Used: 18

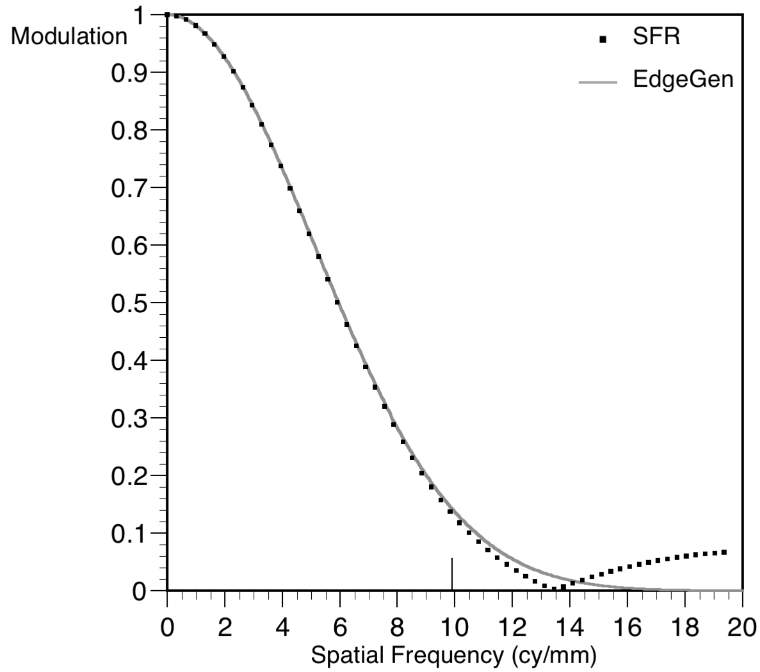
Case: Sharpened Edge



EdgeGen:
Max/Min Gray: 255/0
Noise: 0
Tilt Angle: -5.19443^o
ppi: 500
pixel width: 0.0508 mm
Nyquist: 9.84 cy/mm
MTF type: Sharpened Edge (ksharp=0.12 vsharp=0.15 Gaussian base fc=8.1)
OECF: linear

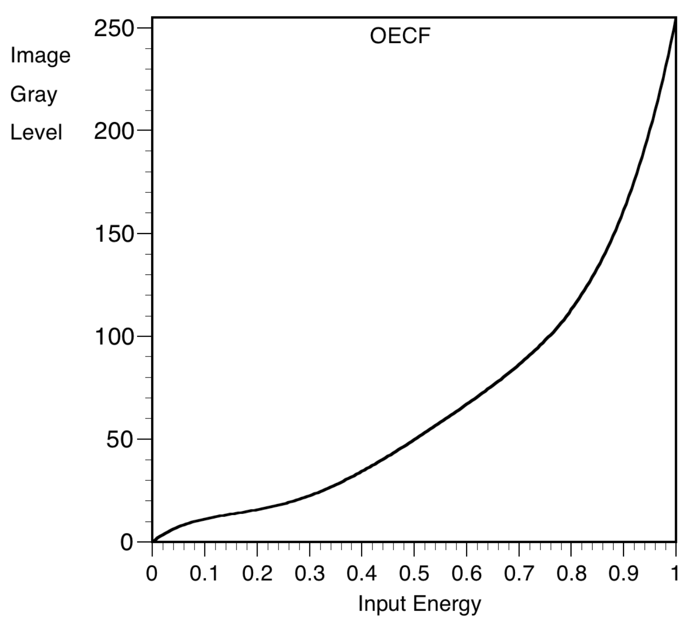
MITRE_SFR:
ROI width: 60
ROI height: 100
Calculated Tilt Angle: -5.194^o
# of Phase Cycles Used: 9

Case: Nonlinear OECF

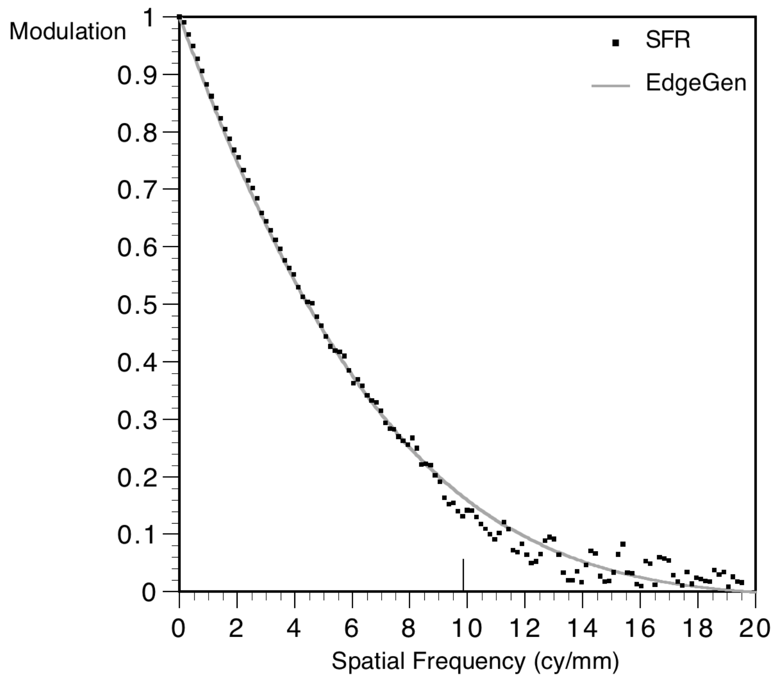


EdgeGen:
Max/Min Gray: 255/0
Noise: 0
Tilt Angle: -5.19443°
ppi: 500
pixel width: 0.0508 mm
Nyquist: 9.84 cy/mm
MTF type: Gaussian (fc=8.1)
OECF: nonlinear

MITRE_SFR:
ROI width: 60
ROI height: 110
Calculated Tilt Angle: -5.194°
# of Phase Cycles Used: 10



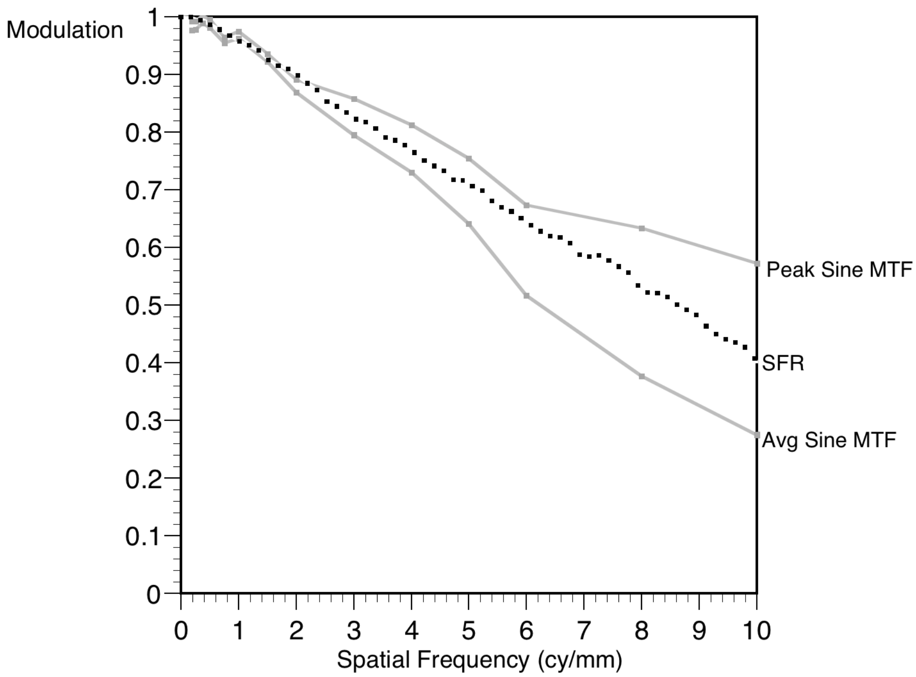
Case: Noisy Edge



EdgeGen:
Max/Min Gray: 245/10
Noise: 3.5 standev graylevels
Tilt Angle: -5.19443^o
ppi: 500
pixel width: 0.0508 mm
Nyquist: 9.84 cy/mm
MTF type: NegExp (fc=7.3)
OECF: linear

MITRE_SFR:
ROI width: 124
ROI height: 200
Calculated Tilt Angle: -5.167^o
# of Phase Cycles Used: 18

Case: Edge vs. Sine Wave (Real Scanner)



Sine wave target and edge target scanned together on flatbed scanner set to 500 ppi with linear OECF. MITRE's *sinemtf* program applied to sine image, which computes peak MTF (from best-phase sine period) and average MTF (from average-phase across all sine periods) of the scanner. "SFR" in plot is output of MITRE_SFR program divided by MTF of edge target. The edge target MTF was computed by scanning the edge and sine targets together at 2500 ppi (true optical resolution) and dividing edge SFR output by *sinemtf* peak output.

MITRE_SFR:
ROI width: 116
ROI height: 236
Calculated Tilt Angle: -5.693^{o}
# of Phase Cycles Used: 23
500 ppi (9.84 cy/mm Nyquist)

References

1. ISO 12233:2000, “Photography – Electronic still-picture cameras – Resolution measurements,” First Edition, 2000-09-01

2. ISO 12233 Slant Edge Analysis Tool sfrmat 2.0, http://www.i3a.org/zip/sfr2.zip

3. 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.

4. FBI specification for single finger capture device for Personal Identity Verification (PIV), http://www.fbi.gov/hq/cjisd/iafis/piv/pivspec.pdf