Q: Can I use PixelScope to measure the MTF of a CCD camera?

Can I use PixelScope to map the actual sensitive area of a pixel?

A: Yes to both. This note explains how.

Approach

This can be a complicated and confusing subject. I've present the material in very small steps, in the hope that it will be more intuitive. Reader comments and suggestions are welcome.

Background:

A CCD consists of an array of individual detector sites, as suggested by the sketch below:

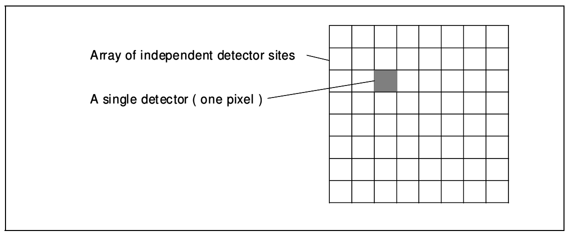

Figure 1: idealized pixel array

The sketch implies that the detectors are independent, and this is generally true: modern CCD arrays have very little electrical crosstalk from one pixel to another. The sketch also implies that the detector is a perfect square whose sensitive area completely fills the pixel area, but this is not generally true. The electronic circuits necessary to read out electrons from each cell occupy a significant portion of the cell area[1]. Pixel MTF and fill factor are related, as this note will demonstrate.

Mapping the sensitive area of a single pixel

We can use a collimator+microscope lens to project an image of a very narrow line onto the CCD. Eventually we'll have to take diffraction into account, but for now let's assume we the width of the line is negligible compared to the width of the pixel

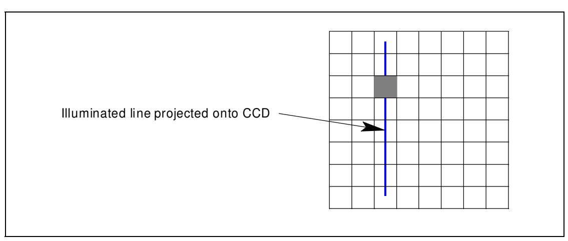

Figure3: Illuminating a single column of pixels

Now imagine we slowly move the line across an area a few pixels wide and monitor the output of a single pixel. With this technique we can map out the sensitivity of a single pixel of the CCD.

---

1 Some CCD chips, including the Sony Hyper HAD chips in cameras sold by Wells Research, have a small "lenslet" array over the actual CCD. These capture light over nearly the entire pixel area, and direct it to the smaller sensitive area of the detector. This is illustrated schematically in figure 2. However, even with a lens array the fill-factor is never 100%

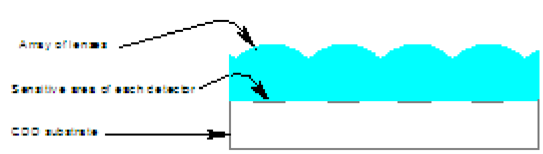

Figure 2: CCD with lens array

For an idealized square pixel the cross section will be a "top hat" as suggested in figure 4.

MTF
The MTF of a "top hat" pixel is easily calculated. It is a sinc function, as show in figure 5.

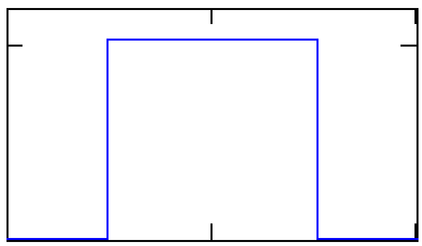

Figure 4: response of idealized pixel to a narrow illuminated line moving across the pixel. Vertical axis is intensity. Horizontal axis is position in um.

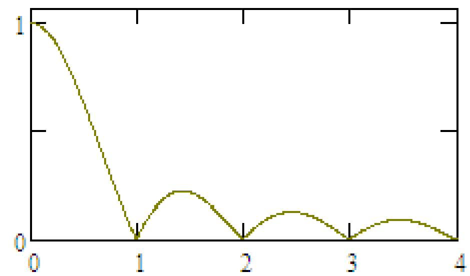

Figure 5: MTF of idealized pixel. Vertical axis is contrast. Horizontal axis is spatial frequency in cycles per pixel.

Note that the MTF goes to zero at 1/pixel pitch. This is a key point. If you are a FFT guru then this behavior may make perfect sense, but for the rest of us it is worth pondering for a moment. Appendix "A" discusses this issue, and explores the effect of pixel shape on MTF.

Practical limitations with the moving line technique
Physically moving a very narrow line across a single is fine in theory, but the pixels on modern cameras are only a few microns wide. In practice it would take a precision motorized stage or a very steady hand to get good data.

However, a technique exists to get the necessary information from a single stationary image. The key is to tip the projected line slightly, as suggested in figure 6. (Compare this to the non-tipped line in figure 3.)

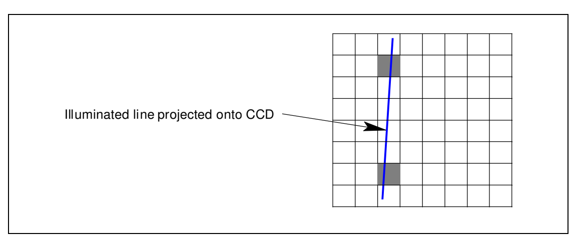

Figure 6: Tipping the illuminated line

Consider the two highlighted pixels in figure 9: The line "samples" these pixels at slightly different locations in the X direction. In effect, we are able to "scan" across the pixel with a single, stationary line.

Real-life example
A 10X microscope lens was used to project a very narrow line[2] onto a Sony CCD with 7.4 micron square pixels.

The image was captured with our PixelScope software, as shown in Figure 7[3]. The cross section of the line is shown in Figure 8.

2 The image of the line is much narrower than the pixel, but because of diffraction its width is not negligible. This issue is discussed in a subsequent section.

3 The blue rectangle was dragged with the mouse. It tells PixelScope what area to display in the cross section plot. Note that the selection rectangle is slightly tipped

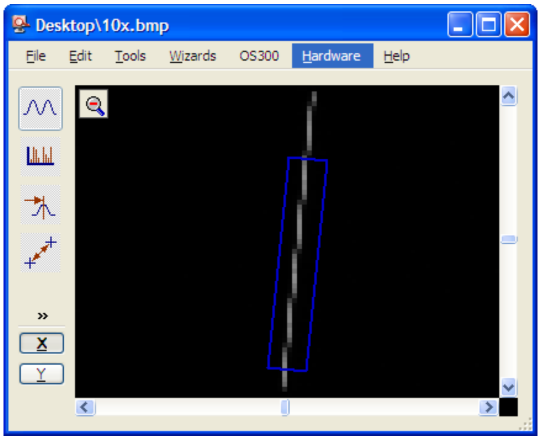

Figure 7: Narrow line projected onto CCD

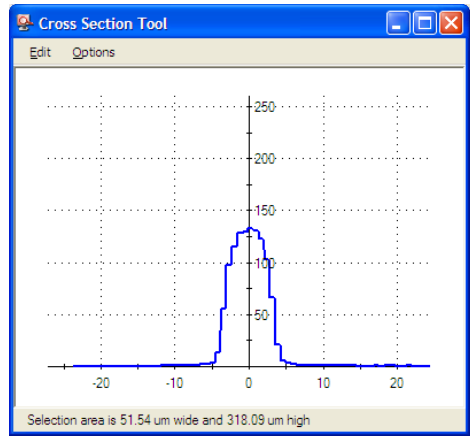

Figure 8: "cross section" plot of line intensity

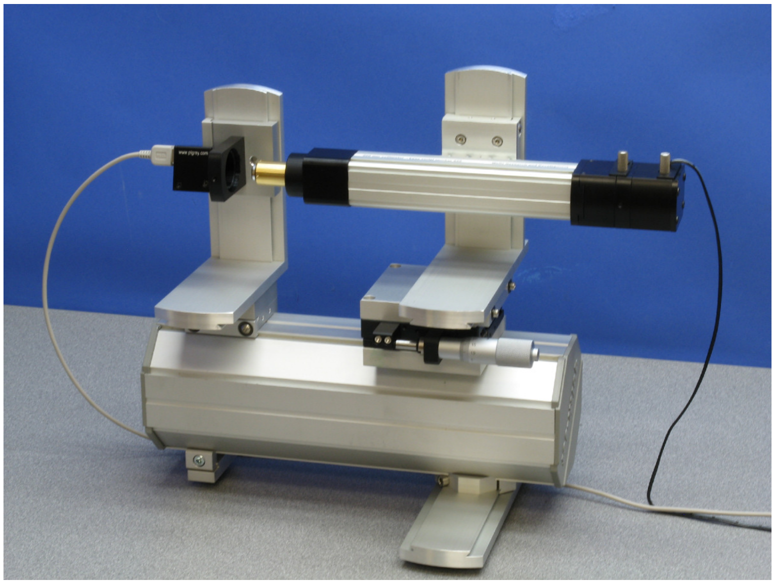

Figure 9: setup used to capture data
CCD camera is at left.

Diffraction effects

The line formed by the 10X lens is narrow, but is still about 1-2 microns wide at the plane of the CCD, This is almost (but not quite!) small enough to ignore. Some versions[4] of PixelScope include an option to correct the MTF plot for the effect of non-zero line width

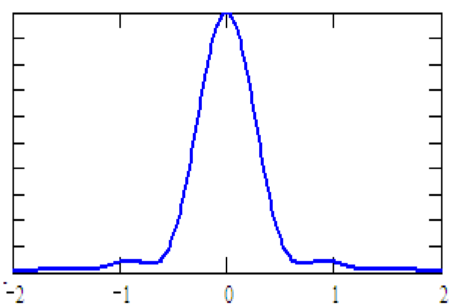

Figure 10: cross section of line projected by 10X microscope lens. (units are um.)

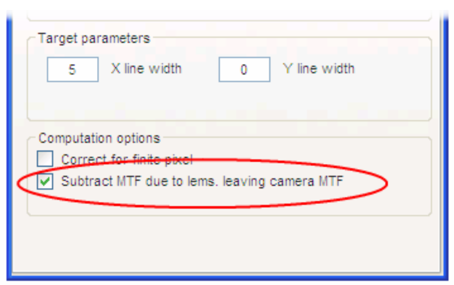

Figure 11: PixelScope can remove effect of line-width before displaying pixel MTF

4 The option is free, but to avoid confusion we only distribute it to customers on request.

MTF
PixelScope's MTF wizard can calculate pixel MTF directly from the cross section shown in figure 8. The result is shown in figure 12.

So how does this compare to the predicted MTF of our idealized square pixel? The match is not very good, as shown in figure 13. One obvious flaw is that the frequency of first zero in the measured data is higher than expected for a pixel with 100% fill factor.

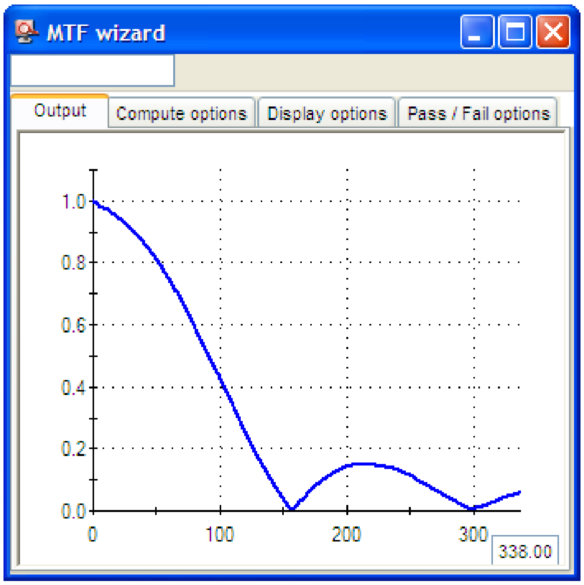

Figure 12: MTF of CCD

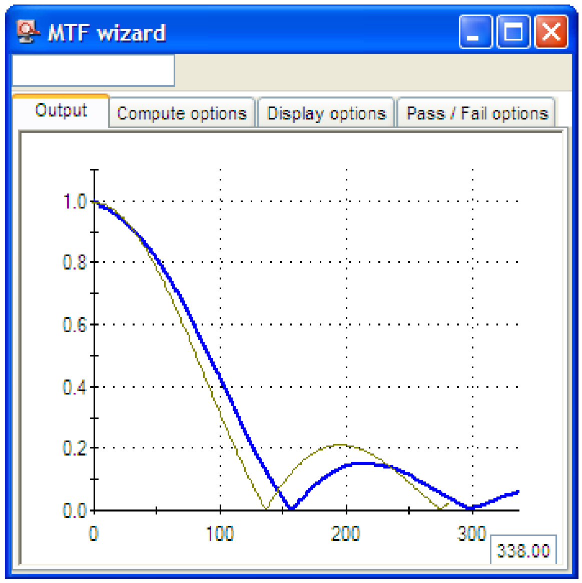

Figure 13: Measured MTF vs theory for idealized pixel with 100% fill factor

However a pixel with 100% fill factor was never realistic anyway. If we model the pixel as an ideal square that fils 87% of the cell area then we get a much better fit:

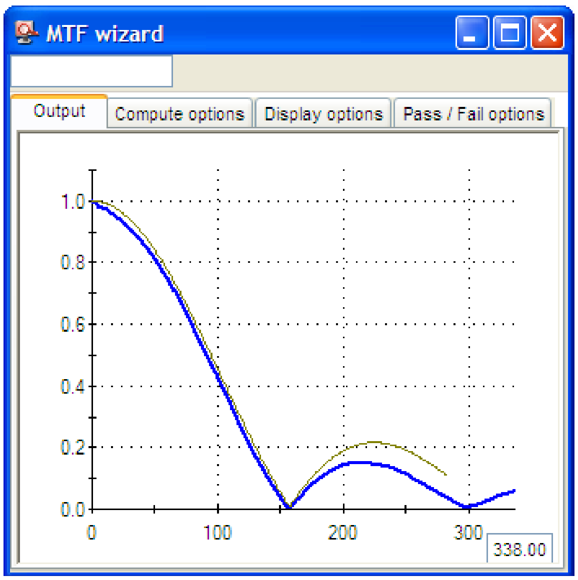

Figure 14: MTF of square pixel with 87% fill factor.

If we are only interested in MTF up to the first zero then this is probably a good enough model.

However, having come this far, I wanted to understand why the peaks following the first zero were not as high as theory predicted. I used MathCAD to explore the effect of various pixel geometries. (See Appendix “A” for more details.) Here is a summary:

- Changing the width of the active area changes the location of the first zero
- Cutting off the corners of the pixel changes the height of the peak following the first zero.

These are not completely independent: cutting off the corners also shifts the first zero a bit.

A perfect fit

After a little experimentation in MathCAD, I concluded the pixel shown in figure 15 was consistent with the observed MTF. The discrepancy is barely noticeable.

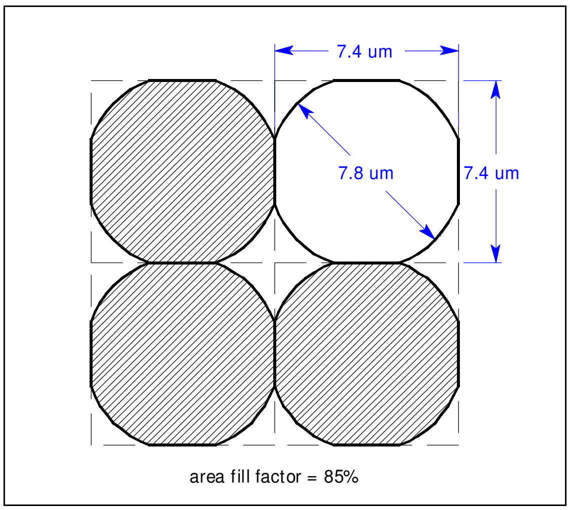

Figure 15: Calculated pixel shape that would produce MTF shown in Fig 16.

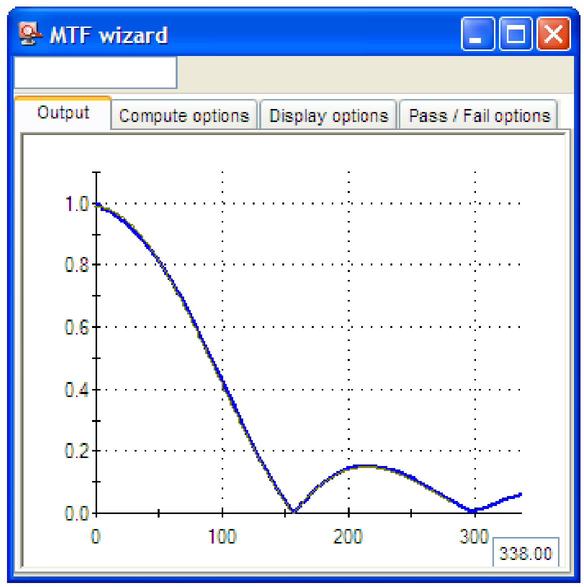

Figure 16: calculated MTF for fig. 14 pixel (red) vs. measured MTF (blue)

In summary:

- The tipped line technique makes it possible to measure Pixel MTF from a single image
- It is possible to deduce the size and shape of the pixel active area from the MTF curve

Appendix A: Pixel cross section and MTF

1: Why does the MTF have "zeros"

The MTF of an idealized "top hat" pixel is a sinc fuction (sin(x)/x) as shown below. The function has zero value at 1/pixel pitch. This section discusses why this happens.

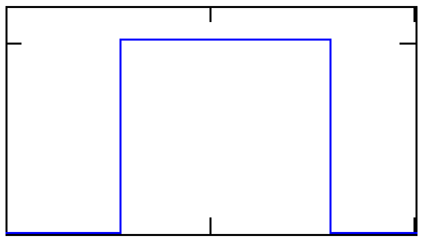

Figure A1: Cross section of idealized square pixel

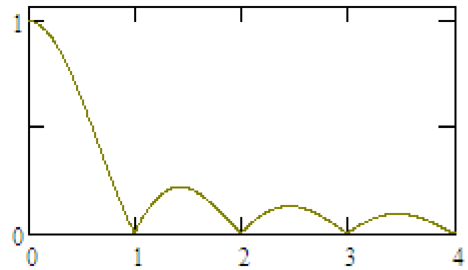

Figure A2: MTF of pixel in figure A1

Figure A3 is often presented to explain the existence of the zero in the MTF curve. The square represents a pixel, and the sine wave represents the intensity of a sine wave target with period= 1/pixel. The output of this pixel will be the convolution of the sine wave and the pixel.

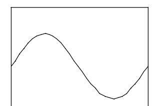

Figure A3

A little thought will show that the output does not depend on the location (phase) of the sine wave. One full cycle will always cover the pixel, and the output of the pixel will always be the same. Therefore the pixel cannot detect the presence (or absence) of this sine wave, therefore the MTF is zero at this frequency

2: If the cross section is not a top-hat will the MTF still have zeros?

A perfect "top hat" cross section is fine in theory, but what if the sensitivity vs. position looks like Figure A4?

Will the MTF still have a zero? The answer is yes. I find the following appeal to intuition helpful:

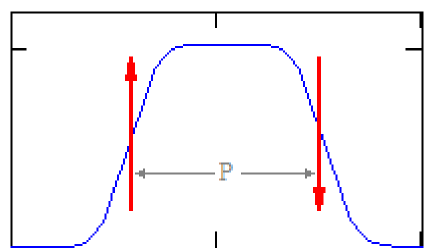

Figure A4: Pixel whose sensitivity is not a "top hat"

- MTF is mathematically linear, in the sense that it obeys superposition. Thus the MTF of the sum of two waveforms will be equal to the sum of the MTF of the individual waveforms.
- Now consider the rising and falling edges to be separate functions, as suggested by the red arrows. From superposition together they make the blue curve. They are identical signals, separated in space by a distance P, and opposite in sign.
- Let's compute the frequency content of the blue curve at a specific frequency, namely 1/P To do this we multiply the signal by a sine wave with period P. Because of superposition we could just as well do this with the rising edge and falling edge separately, and sum the results when we finished. However, it turns out that at frequency 1/P the two sums exactly cancel. This happens because the two signals are of opposite sign, and separated by P. QED the MTF is zero at 1/P

This is not mathematically rigorous (at least not in the form presented) but I find it a helpful way to visualize the math. To me it makes the existence of the zero at 1/P much more intuitive.

Incidentally the sketch suggests a clever way to measure line width: Take the MTF of the image of a narrow line and observe the location of the first zero. This method of measurement is inherently accurate and relatively insensitive to small amounts of misfocus.

3: What if the width of the active area is not the same as the pixel pitch?

It is interesting to note that the pixel cross section in figure 4, and the MTF in figure 5 make no reference to neighboring pixels. What if we keep the pixel pitch the same, but decrease the width of the sensitive area? In other words, the pixel is still a perfect square, the side is only half the pixel pitch.

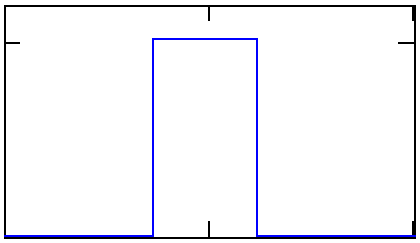

Figure A5. Cross section of a pixel with 50% fill factor. Other factors are same as figure 4,

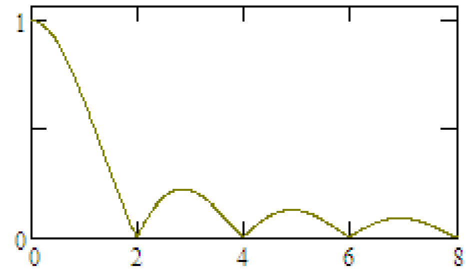

Figure A6: MTF of pixel from figure 6. Note the change in horizontal scale.

The MTF of a top hat is still a sinc function. However, the horizontal scale (which is plotted in terms of pixel pitch) changes by a factor of 2X. The position of the first zero moves out to 2 x 1/pixel pitch.

Again, if you are a FFT guru, this will make perfect sense. As the pixel fill factor goes down, the pixel approaches a perfect "point" sampling device. The MTF of an ideal point sampler is a constant 1 for all frequencies.

To summarize: an idealized square pixel has a sinc MTF curve. If the fill factor is 100%, the first zero occurs at 1/pixel pitch. As the fill factor goes down, the first zero moves to higher spatial frequencies

4: What if the pixel active area is not square? For example, what if it is round?

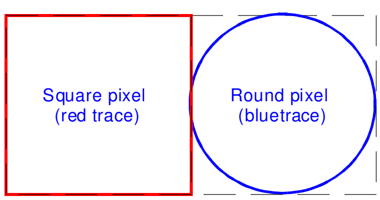

Figure A7: pixel shape

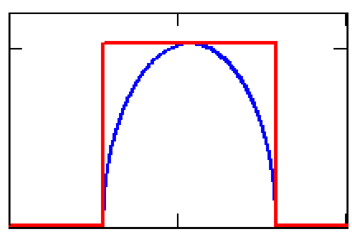

Figure A8: cross section

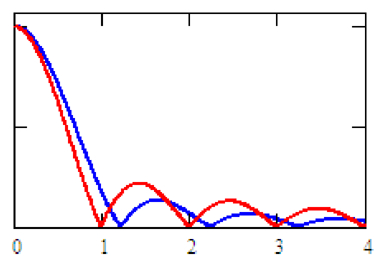

Figure A9: MTF

There are two effects: The location of the first zero moves right, and the height of the subsequent peaks are lower.

5: What if the pixel is some intermediate shape, such as shown in figure x

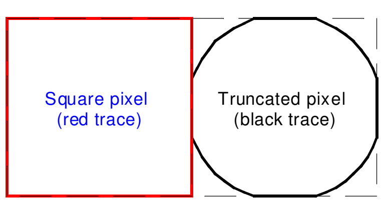

Figure A10: pixel shape

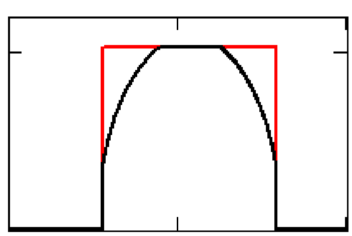

Figure A11: cross section

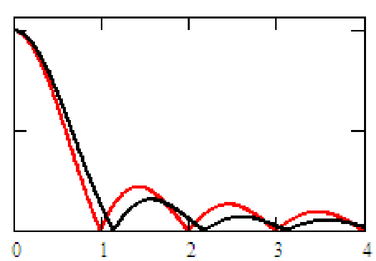

Figure A12: MTF

Perhaps not surprisingly, the MTF is intermediate between a perfect square and perfect round pixel.

6: What if the pixel sensitive area is rotated slightly?

(or conversely, what if we measure the pixel by scanning at a slight angle.)

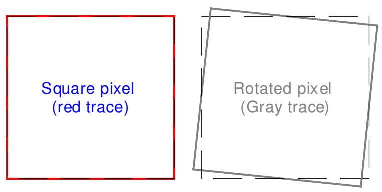

Figure A13 pixel shape

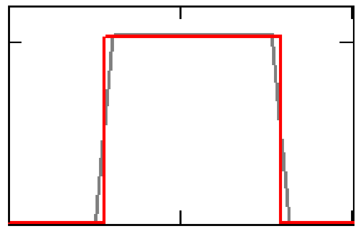

Figure A14: cross section

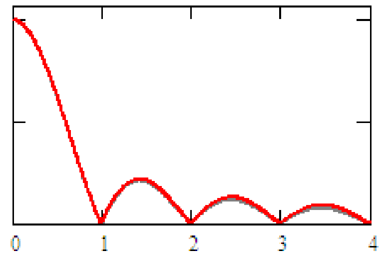

Figure A15: MTF

The MTF is almost identical to the non-rotated pixel.

7: Takeaway conclusions:

- For many pixel shapes the MTF looks roughly like a sinc function.
- Making the pixel active area smaller (relative to the overall pixel grid) moves the location of the first zero to the right.
- Rounding the corners of pixel does not have much effect on the MTF up to the first zero, but does lower the height of subsequent peaks.