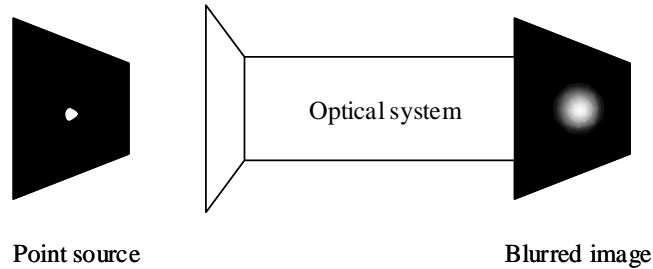


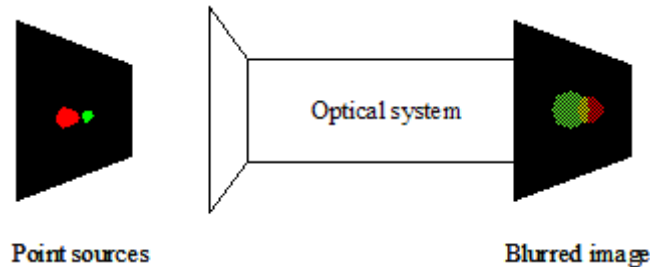
Point-spread and modulation-transfer

“Write about half a page explaining the concept of Modulation Transfer Function (MTF). Discuss its relationship to the ‘impulse’ or ‘point spread’ function (PSF). Include sketches to illustrate your points. Find 2 or 3 examples of real MTFs (e.g. from a camera lens manufacturer’s web page) and compare and contrast your examples.”



When a point source is imaged through some optical system, it may be spread across several points on the image sensor (not necessarily a contiguous group either). The point-spread function describes this lack of one-to-one correspondence; it represents the image that would be formed by an ideal point source.

When a more complex object is imaged (more than just a point-source), the visual cross-section of the object may be treated as a collection of point sources. Each point source element will illuminate several points on the image sensor, resulting in a linear superposition of the point-spread function (with varying spatial offsets) being recorded.



While the point-spread function allows the image to be calculated with ease from the object (by a convolution) it does not provide a nice way for this effect to be reversed (assuming that a system of several million/billion linear equations to solve is not what you would consider “nice”).

The convolution can be calculated by an integral, if the source and point-spread function are both described by known mathematical functions. Otherwise, superimposing a point spread on the image for every point source in the object would be very time-consuming (1000x1000 greyscale image and PSF $\rightarrow 10^{12}$ addition operations to generate image – over five minutes on the average home PC). An easier solution would be to apply the convolution theorem $[f * g = \mathbb{F}^{-1}(\mathbb{F}(f) \times \mathbb{F}(g))]$, which reduces the operation to a pointwise multiplication and some Fourier transforms (which may be implemented by an algorithm such as FFT). For a 1000x1000 greyscale image and PSF, this would involve two FFTs and 10^6 multiplication operations – which would take around a second on a typical PC.

While simulating the effect of an imperfect optical system on a computer would generally not be of much use, reversing such an effect with a computer is, as it allows the imperfections of the

system to be negated. Being able to reverse such effects with a FFTs and pointwise multiplication allows the imperfections to be removed without the need to invert a massive tensor (corresponding to the PSF) – instead, by simply dividing the Fourier transform of the image by the Fourier transform of the point-spread function (deconvolution), then applying the inverse transform.

For example, a system with a Gaussian blur PSF will have a Gaussian MTF (Fourier self-pair). This MTF may be interpreted as an effect that attenuates higher spatial frequencies more than lower ones – so “sharp” details will be diminished by it, while “smoother” details (low frequencies) will pass through almost unaltered:

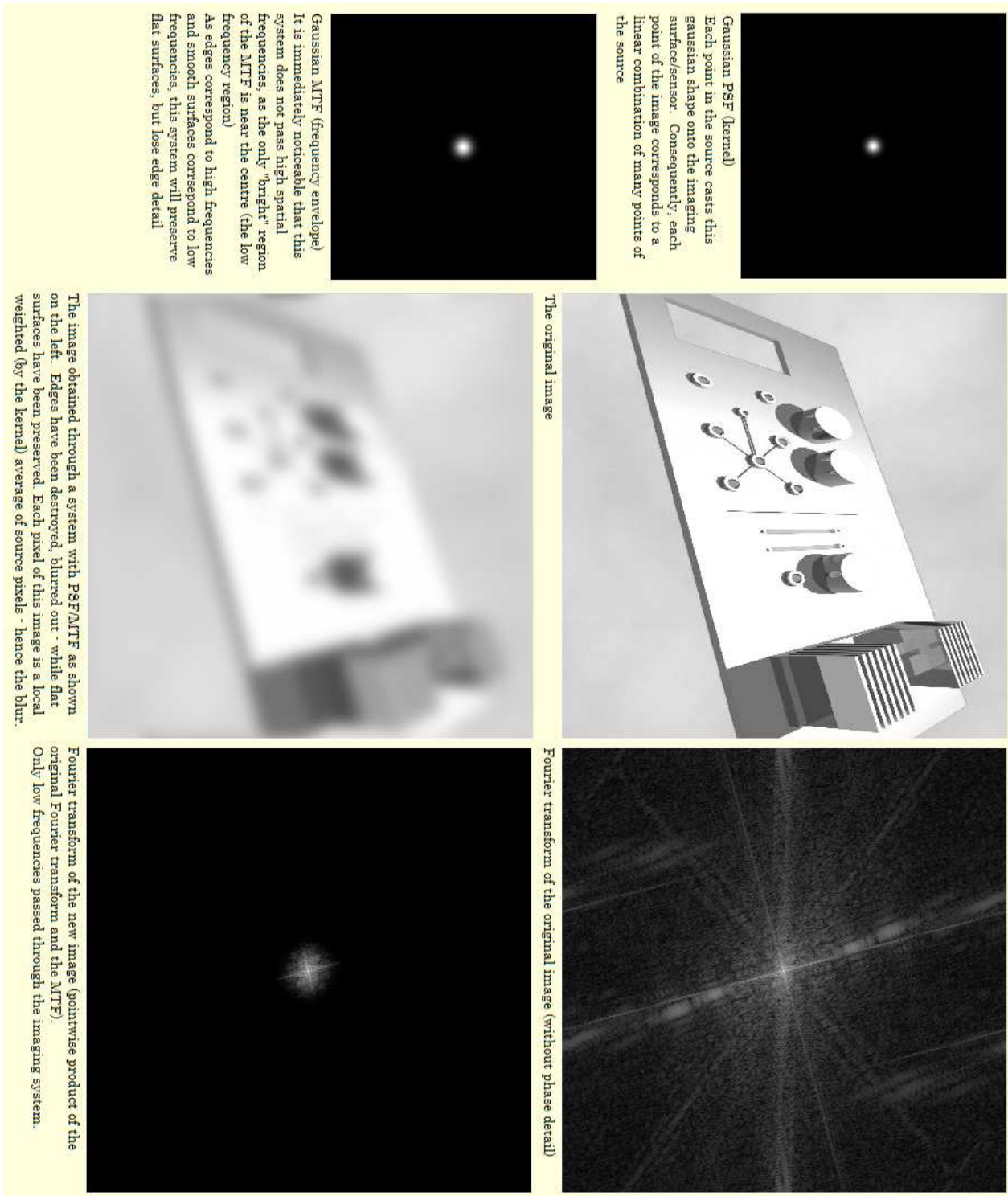
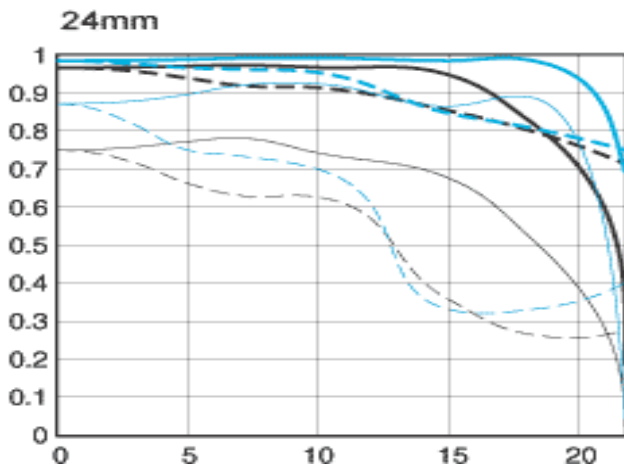


Figure 1: Effect of a large Gaussian PSF on an image and its Fourier transform

Generated by Holo available on request from www.battlesnake.co.uk - source image: Personal CAD project

Canon 2.8/24mm (EF 24-70mm f/2.8L USM)
 £900 online RRP: \$1400

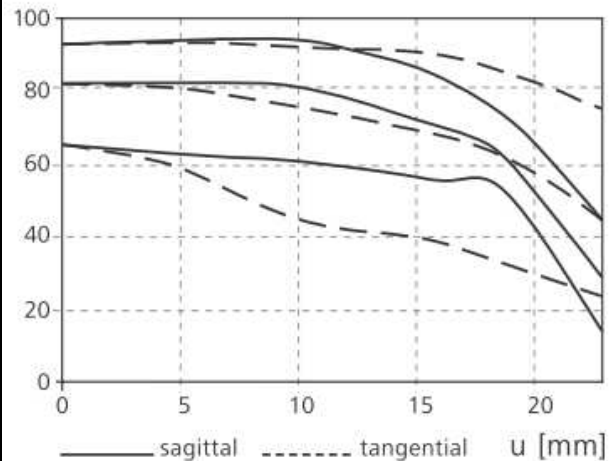
Carl-Zeiss 2.8/25mm (Distagon T* 2.8/25mm ZF)
 £700 online RRP: Unknown



http://consumer.usa.canon.com/cusa/consumer/products/cameras/ef_lens_lineup/ef_24_70mm_f_2_8l_usm

- X-axis is distance from centre of frame
 - Solid: Radial; Dashed: Tangential;
 - Thick: 10 lines/mm; Thin: 30 lines/mm;
 - Black: f/2.8; Blue: f/8;

MTF [%], f-number: k = 2.8



[http://www.zeiss.com/C12567A8003B8B6F/EmbedTitelIntern/D_2.8_25ZF_EN/\\$File/D_2.8_25ZF_EN.pdf](http://www.zeiss.com/C12567A8003B8B6F/EmbedTitelIntern/D_2.8_25ZF_EN/$File/D_2.8_25ZF_EN.pdf)

- X-axis is distance from centre of frame
 - Solid: Radial; Dashed: Tangential;
 - Series: 10, 20, 40 lines/mm (top to bottom)
 - All at f/2.8

At a glance, the Canon would seem like the obvious winner here – with an almost flat response (> 80%) to a line spacing of 10 mm^{-1} across most of the frame (both f/2.8 and f/8 series). The Zeiss performs better tangentially at this line spacing, at f/2.8, however an excess of 10% in visibility beyond 15 mm from the frame centre for the Zeiss is compromised by a 10% shortfall in radial visibility at this region of the frame.

At 30 mm^{-1} line spacing, the Zeiss performs similarly to the Canon radially (interpolating the 20 mm^{-1} and 40 mm^{-1} series for the Zeiss) until around 15 mm from frame centre where the Canon loses high-frequency sensitivity rapidly. The Zeiss is comparable to the Canon tangentially at this frequency, until approximately 13 mm from the frame centre, where the Canon loses sensitivity rapidly (and becomes comparable to the Zeiss at 40 mm^{-1} line spacing).

Over all, the two lenses seem quite closely matched, with the Zeiss providing sharper details towards the edge of the frame and the Canon being sharper towards the centre. Choosing between them purely on MTF would be a question of whether you wanted consistent sharpness across the image (Zeiss) or a sharper centre, but less detailed surroundings (Canon).

A camera enthusiast comparing MTF charts would probably find this a tough decision, without knowledge of how the charts are obtained. The deviation of tangential from radial MTF is a measure of stigmatism, for which the Zeiss outperforms the Canon. High astigmatism incurs a compromise between horizontal and vertical sharpness, commonly resulting in highly directional blurring of the image. Lower astigmatism results in a more directionally isotropic sharpness.

Zeiss measures their lenses with a white-light source, producing graphs that are representative of real-world use of the lenses, however Canon produces their graphs from computer simulations, which (for other manufacturers) have been known to imply lenses that could image beyond the diffraction limit – obvious inaccuracy for far-field imaging!