1 mtf_generate_rectangle

mtf_generate_rectangle — generate images containing edges with known MTF50 values

Synopsis

mtf_generate_rectangle [*OPTIONS*]

DESCRIPTION

mtf_generate_rectangle renders an image (default rect.png) containing a rectangle rotated through the specified angle; other polygonal objects can also be rendered. The image is generated with a specified type of point spread function (PSF), which can either be a simple Gaussian, or a diffraction pattern with an optional optical low-pass filter. Various parameters can be adjusted to produce the desired size / shape of the PSF. Sensor noise is simulated using either a simple additive Gaussian model, or a more sophisticated sensor noise model. The program reports the analytical MTF50 value of the rendered edges. **mtf_generate_rectangle** can be used to evaluate the accuracy of the **mtf_mapper** tool, since the accuracy of the images generated with **mtf_generate_rectangle** have been validated by third parties.

OPTIONS

--b16

>   Render output image as a linear intensity 16-bit grayscale image (default is 8 bits gamma corrected)

-l

>   Render output image as a linear grayscale image (default is sRGB gamma)

-d *pixels*

>   Dimension of the long edge of the rectangle, in pixels. Please note that MTF Mapper may produce inaccurate results if this parameter is below 35 pixels.

-x *pixels*

>   Sub-pixel offset applied to rendered rectangle, x-axis direction.

-y *pixels*

>   Sub-pixel offset applied to rendered rectangle, y-axis direction.

-b *std. dev*

>   Specify the standard deviation of the Gaussian point spread function (PSF) with which the edges are sampled. This parameter effectively controls the resulting MTF50 value of the edges. Mutually exclusive with **-m**.

-m *mtf50*

>   Directly specify the desired MTF50 value. This indirectly controls the Gaussian PSF standard deviation. Mutually exclusive with **-b**.

-a *angle*

>   Specify the angle (in degrees) at which the rectangle is generated. Useful for characterising MTF measurement accuracy as a function of edge orientation

-r *ratio*

>   Specify the aspect ratio of the rectangle in the range [0,1]. Ratios above 1.0 are clipped. The aspect ratio is adjusted so that the rectange is at least one pixel wide. Very thin rectangles (< 25 pixels wide) are not ideal for use with MTF Mapper.

--esf-only

>   Render only a cross-section across the edge of the rectangle at a 1/32th pixel spacing. This is mainly useful for performing synthetic image accuracy measurements, but can be used to generate Point Spread Functions (PSFs) for the *box* and *4dot-olpf* Airy pattern variants (these PSFs are hard to compute analytically). Default output filename is *profile.txt*, but can be specified with **-o**.

-o *filename*
>   Specify the name of the output file

-h
>   Displays usage information

NOISE OPTIONS

-n *std. dev*
>   Specify the noise standard deviation [0-1) of the Gaussian noise that is added to the image. This invokes the simplified noise model (which is also the default), which produces noise that is independent of signal level. A more accurate sensor noise model can be obtained with the full sensor noise model (**--read-noise**, etc.); See examples below.

--adc-depth *bits*
>   Number of bits used in full noise model ADC. This parameter effectively scales photon shot noise. Recommended value is 12.

--adc-gain *electrons*
>   Number of electrons per DN, i.e., the ADC gain of the full noise model. Recommended value is 2.64.

--pattern-noise *fraction*
>   Magnitude of Pixel Response Non-Uniformity (PRNU) component of full noise model. Recommended value is 0.0085.

--read-noise *std. dev*
>   Standard deviation of read noise component of full noise model, in electrons. Recommended value is 3.7.

-c *factor*
>   Reduce edge contrast by *factor*, i.e., if *factor* is 0.3, then the edge contrast is 1 - 0.3 = 0.7.

-s *seed*
>   Specify the random seed for the RNG used to generate the additive Gaussian noise.

PSF/RENDERING OPTIONS

-p *gaussian|gaussian-sampled|airy|airy-box|airy-4dot-olpf*
>   Specify the desired PSF type. PSF types *gaussian* and *gaussian-sampled* accept the **-m** and **-b** arguments to control PSF shape, and produce, unsurprisingly, Gaussian-shaped PSFs. Note that these PSFs are pure Gaussians, i.e., they have not been convolved with the photosite aperture, and are therefore not (directly) useful for sensor simulations. The remaining PSFs, namely *airy*, *airy-box* and *airy-4dot-olpf* simulate an Airy pattern PSF, such as would be observed when diffraction through a circular aperture is considered. These accept the **--lambda**, **--pixel-pitch** and **--aperture** arguments to control the diffraction pattern. The *airy* PSF models only diffraction, and does not include convolution with a photosite aperture. In contrast, the *airy-box* PSF includes this final convolution step, and is thus a useful PSF for simulating lens + sensor combinations that do not include an Optical Low-Pass Filter (OLPF), also commonly called an Anti-Aliasing (AA) filter. Lastly, the *airy-4dot-olpf* PSF simulates a lens + OLPF + sensor combination, assuming that the AA filter is a 4-way beam splitter (e.g., Lithium Niobate birefringent designs) with a displacement of 0.35 pixels, which appears to match current Nikon DSLR sensor designs.

--airy-samples *n*
>   Number of half-samples (n) per axis per pixel for Airy PSFs (actual number of samples is (2n+1)^2).

--olpf-offset *pixels*
>   4-dot OLPF beam splitter offset, default value is 0.35.

--lambda *micron*
>   Specify the wavelength of the light to simulate when rendering diffraction PSFs, in micron.

--pixel-pitch *micron*
>   Specify the photosite pitch of the sensor. This is required to convert the wavelength parameter into a pixel-scale value.

--aperture *f-number*
> Specify the relative numeric aperture of the lens used to render a diffraction PSF.

--target-poly *filename*
> Target polygon data file. This option overrides the default target (a rectangle) with the geometry of the polygons specified in `filename`. The format of this file is just

```
_number_of_vertices_
_x1_ _y1_
_x2_ _y2_
...
```

> which can be repeated for additional polygons. The polygons should be simple, but may be concave, and should have fewer than 400 vertices. The default ROI will range from (0,0) up to the extent of the vertices; see the **--roi-** options to override this behaviour.

--photosite-geom *square|circle|rounded-square*
> Photosite aperture geometry, default is *square*. The *circle* and *rounded-square* are 64-sided polygonal aproximations. If you require a more general aperture geometry, use the **--photosite-poly** option.

--photosite-poly *filename*
> Photosite aperture polygon file name. The origin is (0,0), with a 100% fill-factor square pixel defined as the region (-0.5, -0.5) to (0.5, 0.5). The format of *filename* is the same as that of **--target-poly**.

--roi-height *pixels*, **--roi-width** *pixels*, **--roi-col** *pixels*, **--roi-row** *pixels*
> Specify the Region Of Interest (ROI) that should be rendered, the region (row, col) to (row + height, col + width). This is useful for generating a fixed-size output image, or just rendering a subset of a larger geometry (see **--target-poly**).

--fill-factor *factor*
> Fill-factor of photosite [0.01,1], default is 1.0.

EXAMPLES AND NOTES

This utility supports both a simple Gaussian noise model, and a signal-dependent noise model that models sensor noise more accurately. If you just need *some* noise, then the simple model (specified with the **-n** parameter) will be sufficient. This method can also be used to generate noise-free images.

Noise free images
> mtf_generate_rectangle -n 0 . . .

Simple noise model
> mtf_generate_rectangle -n 0.01 . . .
> #Additive Gaussian white noise with a standard deviation of 1%. This is the default option.

Full noise model
> mtf_generate_rectangle --pattern-noise 0.0085 --read-noise 3.7 --adc-gain 2.64 --adc-depth 12 -c 0.2
> #Realistic simulation of Nikon D7000 noise.

Dynamic range and headroom
> The default **-c** parameter setting (0.1) will cause the edge to span the range [0.05, 0.95], leaving some dynamic range for subsequent processing steps. For example, the additive noise will be clipped if there is no headroom for the outer tails of the noise distribution. The headroom is also criticial for other types of image processing, e.g., sharpening or resizing. When in doubt, rather specify some extra headroom (**-c** 0.2), especially if you push the noise levels higher.

Linear vs. Gamma corrected values By default, mtf_generate_rectangle will generate output files with linear intensities only when 16-bit output is specified (**--b16**). This implies that if no parameters are specified, then **mtf_generate_rectangle** will generate 8-bit files in the SRGB space, i.e., intensities will be gamma corrected. You can override this behaviour with the **-l** switch, which will force linear 8-bit values. It is crucial that you keep track of whether you are dealing with gamma-corrected intensities or not. For example, **mtf_mapper** will not measure correct MTF values if there is a gamma mismatch (i.e., generated

with linear 8-bit intensities but processed as if images were gamma corrected). Please use the **-l** switch with **mtf_mapper** if you generated your test images with the **-l** switch of **mtf_generate_rectangle**.

Better yet, always use 16-bit images, since all the MTF Mapper tools assume they are linear.