On the rendering of synthetic images with specific point spread functions

F. van den Bergh

Remote Sensing Research Unit, Meraka Institute
CSIR, PO Box 395, Pretoria
South Africa, 0001
fvdbergh@csir.co.za

***Abstract*—Most image processing and machine vision algorithms are evaluated on synthetic images, usually of known target patterns, to determine their effectiveness under controlled conditions. Such synthetic images are often rendered using an area-weighted strategy, which implies that the point spread function (PSF) of the simulated optical system is a box function. This paper discusses several rendering strategies that can be employed to extend the generation of synthetic images to more general point spread functions. In particular, high-accuracy algorithms for rendering Gaussian and circular aperture diffraction PSFs are presented.**

I. Introduction

Machine vision algorithms are typically hard to implement correctly because small errors in the implementation may not necessarily lead to easily observable errors in the output. To guard against such implementation errors it is prudent to test the algorithm under controlled conditions. This usually requires synthetic images with known properties, such as dynamic range, signal-to-noise ratio, noise distribution, and optical system point spread function.

A quick review of the literature will reveal that many such experiments simplify the synthetic image generation process by assuming that the noise is additive in nature, with a Gaussian distribution, and that the optical system PSF can be approximated as a Gaussian. These assumptions make for an efficient implementation, especially if a Gaussian blur is used to simulate the effect of the PSF at the target resolution of the synthetic image. Although these assumptions are not inherently poor for the evaluation of many machine vision algorithms, it is desirable to have more realistic simulation methods available for evaluating those methods that require greater accuracy in PSF and noise simulation.

A few algorithms can be tested rigorously using relatively simple test images. One example is the slanted edge algorithm that estimates optical system resolution by computing the Modulation Transfer Function (MTF) of a knife-edge target [1]. For this algorithm the synthetic image can be a simple step function in intensity, with the edge rendered at a specific angle, and with a known PSF. Other examples include super-resolution methods, where multiple low-resolution images are combined to construct a higher resolution image [2]. These algorithms can be evaluated by presenting them with synthetic images of simple geometric shapes (e.g., black polygons on white backgrounds), and measuring the resolution of the super-resolved output using the slanted edge algorithm mentioned above. Lastly, the accuracy of algorithms designed to extract simple features from images, such as rectangle-, circle- or ellipse-detection algorithms [3] can be evaluated on simple synthetic images consisting of black polygons on white backgrounds.

In all of the above cases the algorithms are best evaluated on synthetic images where the PSF closely matches the PSF of the expected real-world application, which typically requires modeling at least lens aperture diffraction and photosite aperture effects. Surprisingly, details on rendering synthetic images with PSFs that accurately capture the desired properties are not often included in papers relying on such synthetic images for validation experiments.

This paper discusses several algorithms that may be used to render synthetic images with specific point spread functions, focusing on some common PSFs, including a box function PSF, a Gaussian PSF, a circular aperture diffraction PSF, and a birefringent crystal optical low pass filter PSF.

II. Background

A. Point spread functions

The *point spread function* describes the finite impulse response of an imaging system, in other words, the response when imaging a point source. The PSF is defined in the spatial domain, and has a frequency domain analogue that is called the *modulation transfer function* (MTF), which can be obtained via the Fourier transform.

Unless the PSF is itself an impulse function, it will distribute the light originating from a point source over a region with non-zero area. Visually, this spreading is perceived as blur; for a point source, we will observe a larger blob. If the true object being imaged is not a point source, the effect of the PSF may be more complex in appearance. In practice, the interaction of the PSF and the discrete sensing elements (photosites) of a digital image sensor can be thought of as "placing the PSF at the centre of each photosite, and weighting the light coming from the true object according to the PSF". The resulting image is thus the convolution of impulse functions placed at the photosite centres, the PSF, and the true object.

If the PSF is *shift invariant*, i.e., identical across the focal plane, then the convolution can be implemented efficiently in the Fourier domain as the product of the Fourier transform of the true object and the MTF, followed by an inverse Fourier transform to return to the spatial domain.

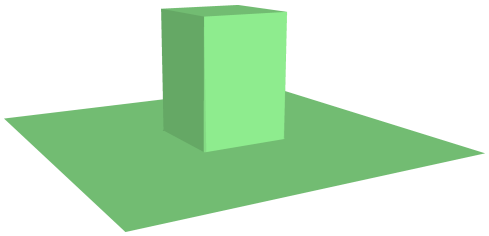

Fig. 1. Box function PSF

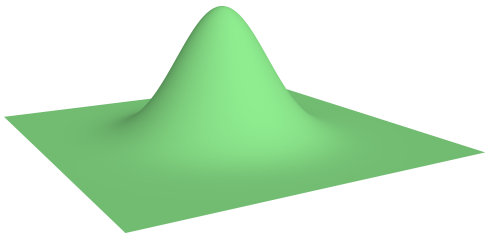

Fig. 2. Gaussian PSF

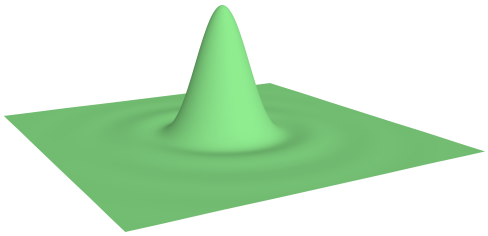

Fig. 3. Circular aperture diffraction PSF

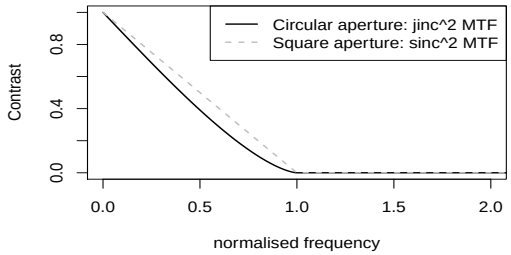

Fig. 4. Circular aperture diffraction MTF

B. Common point spread functions

Some of the commonly occurring point spread functions follow:

1) Box PSF: The box function PSF corresponds to the rectangular photosites of a matrix sensor. As such, any image formed by a matrix sensor will ultimately involve a convolution step with the box function. A fill factor of less than 100% may imply that the effective box function is narrower than the photosite pitch, and a non-square photosite geometry (e.g., L-shaped) may be in effect; both these factors can be modelled if desired by piecewise decomposition into multiple smaller box functions.

The 2D box function is visualised in Figure 1. Fortunately, the 2D box function PSF lends itself to a highly efficient implementation when rendering polygon shapes: each photosite's intensity is simply proportional to the area of the pixel covered by the target polygon. This intersection can be computed using the Sutherland-Hodgman polygon clipping algorithm [4], for example.

2) Gaussian PSF: The Gaussian PSF is often used to introduce a blur effect into synthetic images. Although the Gaussian PSF does not correspond to any common physical phenomenon, it does serve as a coarse approximation to diffraction effects. The primary reason for its popularity appears to be ease of implementation and use.

A 2D Gaussian PSF is shown in Figure 2. Although a direct implementation of this PSF is straightforward, it is rather more involved to obtain highly accurate synthetic images; one such method is discussed below in Section III-D.

3) Circular aperture diffraction PSF: Light passing through a circular aperture is affected by Fraunhofer diffraction to produce a light intensity distribution known as the Airy pattern [5]. The width of this pattern is inversely proportional to the diameter of the aperture, thus smaller apertures produce wider Airy pattern point spread functions. For incoherent light, the Airy pattern is defined as

$$I_0 \cdot \left(\frac{2J_1(x)}{x} \right)^2 \tag{1}$$

where J_1 is the Bessel function of the first kind, of order one, and I_0 represents the peak intensity. Note that $x = \frac{\pi q}{\lambda N}$, where λ is the wavelength of the light, N is the aperture f-number, and q is the radial distance from the axis passing through the centre of the aperture. This PSF is illustrated in Figure 3. The concentric side-lobes of the pattern are barely discernible after the second cycle, however, the support of the Airy pattern is infinite, and the side-lobes never quite reach zero.

The Fourier transform (or Modulation Transfer Function, MTF) of the Airy pattern is the *Chinese hat* function:

$$\mathrm{chat}(s) = \frac{2}{\pi} \left(\cos^{-1}(s) - s\sqrt{1 - s^2} \right) \tag{2}$$

where $0 < s \leq 1$ represents the *normalised spatial frequency*, which is defined such that $s = \lambda N f$, with f denoting unnormalised frequency. This function is illustrated in Figure 4, which clearly shows that the Airy pattern PSF acts as a low-pass filter.

The Airy pattern is of particular importance to the rendering of synthetic images produced by a lens, since even in the absence of a physical aperture stop the lens itself acts as an aperture. For a wavelength of 550 nm and a photosite pitch of 5 micron, diffraction will reduce system resolution for apertures with an f-number greater than f/5.6. For even smaller photosite pitch values, this maximum allowed f-number must be decreased even further to prevent loss of resolution owing to diffraction.

In the frequency domain, the Airy pattern MTF reaches exactly zero and remains at zero beyond the critical frequency $f = \frac{1}{\lambda N}$. This property is poorly approximated by a Gaussian PSF (which also has a Gaussian MTF), which does not decay quite as rapidly as the Airy pattern MTF. Should one wish to approximate the Airy pattern with a Gaussian regardless of

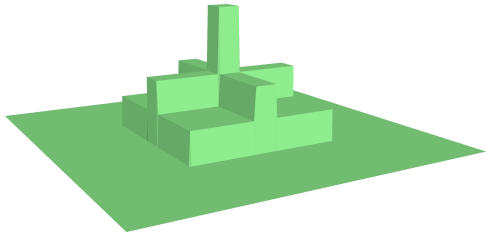

Fig. 5. 4-dot birefringent OLPF PSF with a 0.75 photosite pitch displacement

its limitations, the best-fitting Gaussian approximation in the least-squares sense can be obtained by choosing the standard deviation to be $\sigma \approx 0.425\lambda N$.

4) Birefringent OLPF PSF: An optical low-pass filter (OLPF) can be used to suppress the power at frequencies above the Nyquist limit for a given photosite pitch. Strictly speaking, this is a requirement to ensure correct sampling, and aliasing artifacts may appear in images captured with an optical system that lacks an OLPF. If the lens aperture is chosen carefully with respect to the photosite pitch, it is possible to employ diffraction to act as a low-pass filter, but this approach is not practical for larger photosite pitches (e.g., larger than 5 micron) when used with large relative apertures. If a Bayer colour filter array is included in the sensor design, it becomes even more important to minimise aliasing, which may manifest as colour interpolation errors.

One method of constructing an OLPF is through the use of a birefringent material, i.e., a crystal that forces photons to take different paths depending on their polarization. One such material is Lithium Niobate, which can be used to split an unpolarised beam into one horizontally polarised beam and one displaced parallel beam containing only the vertically polarised photons [6]. If an image passes through such a filter, the resulting image leaving the filter will be the superposition of the image and copy of the image displaced by a distance d, which effectively blurs the image in the direction of the displacement. Two such filters can be combined (with an appropriate depolariser in between) to effect a blur in both directions.

A 4-dot birefringent OLPF PSF is illustrated in Figure 5. The exact shape of the PSF is dependent on the displacement, d, effected by the birefringent plates. In general, it is desirable to choose the displacement as a function of the photosite pitch so that the filter cut-off frequency is related to the Nyquist frequency of the sensor.

An implementation of the 4-dot OLPF for synthetic image rendering is a straightforward extension of the method used for a box function PSF. The process is simply repeated four times with four displaced box function PSFs.

C. Combining point spread functions

As already alluded to above, the system PSF is a combination of the individual PSFs encountered along the optical path. Provided that phase effects can be ignored, such as when light passes from the lens onto the sensor, the PSFs can be combined by direct convolution. Equivalently, the MTFs of the various components along the optical path can simply be multiplied to obtain the system MTF. This approach can be used to combine the lens (diffraction) response, OLPF response (if present) and photosite aperture response to obtain the system response. Unfortunately, the effects of defocus cannot be integrated with this approach, and are therefore not considered in the sequel.

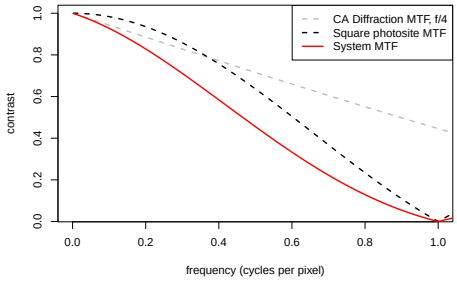

Fig. 6. Diffraction, photosite aperture and combined MTF

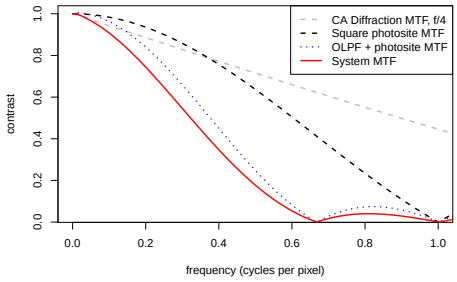

Fig. 7. Diffraction, Photosite aperture, OLPF and combined MTF

Some useful combinations, corresponding to typical configurations encountered in real optical systems, will now be considered.

1) Circular aperture diffraction + photosite aperture: This system corresponds to a monochromatic matrix sensor and lens combination. It is also appropriate for Bayer CFA sensors that do not contain an OLPF. The MTFs of the components, as well as the combined system MTF, are illustrated in Figure 6.

2) Circular aperture diffraction + 4-dot OLPF + photosite aperture: This configuration is common for large-photosite Bayer CFA systems, such as commercial Digital Single Lens Reflex (DSLR) cameras. The OLPF helps to suppress colour interpolation artifacts as well as regular aliasing artifacts. An example of an OLPF MTF curve is shown in Figure 7, using a beam separation distance $d = 0.75$ pixels. This effectively attenuates the system response strongly at frequencies above 0.67 cycles per pixel, but does not completely eliminate power above Nyquist (0.5 cycles per pixel).

III. Rendering strategies

Several strategies for rendering synthetic images will now be discussed. To simplify the discussion, it will be assumed that the target object is a black polygon rendered against a

white background. Furthermore, it is assumed that the edges of the target object are perfect step functions.

One of two basic operations are required to implement the proposed rendering strategies: an indicator function operator, and a polygon-polygon intersection operator. The indicator function operator returns a value of 1 if its argument is inside the target polygon, and 0 otherwise. The polygon-polygon intersection operator returns a real number representing the area of the polygon formed by the intersection of the two polygon arguments to the operator.

Both of these operators can be implemented reasonably efficiently for polygon target objects. A simple point inclusion operator can be defined for some non-polygonal target objects, such as circular and ellipsoidal discs, but these target shapes can be approximated as polygons to the required accuracy if necessary.

All the strategies presented below are attempts to compute the integral that results when convolving the target object indicator function and the desired PSF. Since the extent of the target object is finite, it is convenient to express the intensity of the pixel at location (x, y) in the synthetic image as an integral over the target object, i.e.,

$$I_{(x,y)} = \int_{\mathbb{R}^2} \mathbf{1}_P(\mathbf{x}) f_{(x,y)}(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{3}$$

where $\mathbf{1}_P(\mathbf{x})$ denotes the indicator over polygon P, and $f_{(x,y)}$ represents the PSF centered at location (x, y). This can be simplified to

$$I_{(x,y)} = \int_{P} f_{(x,y)}(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{4}$$

by restricting the integral to the region bounded by the polygon P, when appropriate.

Except for the box function PSF, approximate solutions to these integrals must be obtained using numerical integration methods. When the PSF itself is the result of the convolution of simpler PSFs, e.g., the combined effect of a square photosite aperture and circular aperture diffraction, the problem is compounded because the PSF itself becomes another integral to be approximated. As is often the case, Monte Carlo integration methods are a convenient way of computing these integrals.

A. Uniform oversampling

Using the indicator function, the synthetic image can be rendered by generating a set of sampling points coinciding with the centre of each pixel in the synthetic image. Each of these points can then be tested against the indicator function to determine whether the sample falls inside the target object, or not, colouring the resulting pixel accordingly.

This strategy is computationally efficient, but leads to severe aliasing, visible as "stair steps" along the edges of the target. The aliasing is due to the low sampling rate, at one sample per pixel, compared to the infinite bandwidth required to render the edge correctly. Two straightforward extensions can be employed to mitigate the aliasing: 1) render the synthetic image at a higher resolution, followed by downsampling to the desired resolution, or 2) oversampling on a uniform grid with sub-pixel spacing (Figure 8).

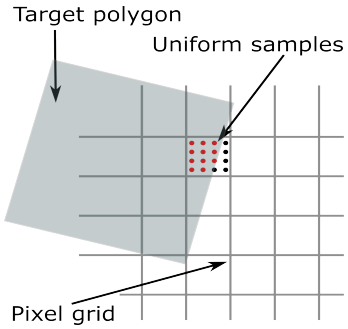

Fig. 8. Uniform oversampling using box PSF indicator function

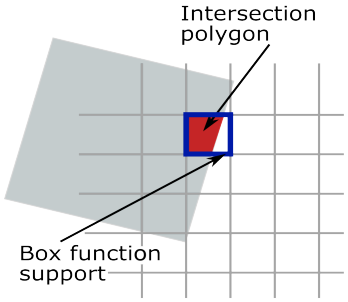

Fig. 9. Area-weighted rendering by polygon intersection

These two oversampling strategies can produce identical results, but the first strategy is computationally more complex, and requires significantly more memory. The additional samples should be weighted according to a properly scaled (spatially) grid of weights representing the desired PSF. Both these strategies introduce distortion of lower frequencies if the PSF is not band limited, i.e., if the support of the PSF is infinite, like in the case of a Gaussian PSF or an Airy pattern PSF. This error is bounded, and clearly an approximation can be constructed at any desired accuracy.

B. Area-weighted sampling

The box function PSF presents a special case for which an exact solution can be obtained efficiently. Note that the support of the box function is finite, with its extent typically being a square with sides equal to the photosite pitch, and that the function is constant over the region where it is non-zero. The result of convolving a box function placed at a given pixel centre and the target polygon is proportional to the area of intersection between the target polygon and the box function's support (Figure 9).

C. Gaussian PSF importance sampling

The Gaussian PSF has infinite support, which implies that any point-based sampling strategy must inherently introduce some error. A naive approach to rendering a synthetic image with a Gaussian PSF would be to use the uniform oversampling strategy (Section III-A), and choosing the individual sample weights from the desired Gaussian function. This strategy has two significant weaknesses: 1) the PSF will be truncated at the boundary of the uniform sampling grid, and 2) the samples that fall in the tails of the Gaussian PSF will

contribute little to the overall integral, yet they outnumber the samples in the central region of the Gaussian where the weights are much larger.

A much better strategy is to compute the convolution integral using Monte Carlo sampling. In particular, *importance sampling* strategies allow us to sample the PSF according to its actual density [7, section 7.6]. If we wish to approximate the integral I over the volume V, then importance sampling reformulates the problem as

$$I \approx \frac{1}{N}\sum_{i=0}^{N-1} f(\mathbf{x}_i) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \tag{5}$$

where $f(x_i)$ represents the integrand, and $p(x_i)$ the probability of sampling point x_i. It is assumed that $\int p(x)\mathrm{d}x = 1$. The benefit of importance sampling is that we can choose a distribution $p(x)$ that is easily invertible, but matches $f(x)$ as closely as possible. Uniform oversampling is simply a special case of importance sampling where all points on the uniform grid are equally likely, and happen to be uniformly spaced.

The sampling strategy is thus to generate a set of sampling positions that follow a chosen distribution $p(x)$, a method known as *inverse transform sampling* [7, section 7.2]. Let F denote the cumulative distribution function of $p(x)$. Then $F^{-1}(U) \sim F$, where U is a uniform variate in the range $[0, 1]$. Thus, starting from a uniform variate u in the range $[0, 1]$, we can obtain a sample x with distribution $p(x)$ by transforming u as $x = F^{-1}(u)$.

This method does not require an analytical form for F^{-1}; a table-based inversion or a polynomial approximation is often adequate. When rendering a Gaussian PSF, we choose to distribute x as $x \sim N(0, \sigma)$, which can be achieved through Moro's inversion [8]. Since we can choose the standard deviation σ to exactly match the desired Gaussian PSF, and generate x with the exact same distribution, we can simplify Equation 5 to

$$\begin{aligned} I &\approx \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)}\mathbf{1}_P(\mathbf{x}_i) \\ &\approx \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{1}_P(\mathbf{x}_i), \end{aligned} \tag{6}$$

since $f(\mathbf{x}_i) = p(\mathbf{x}_i)$. If the sampling distribution of $\mathbf{x}_i$ matches the PSF exactly, then the samples $\mathbf{1}_P(\mathbf{x}_i)$ should not be weighted by the PSF at $\mathbf{x}_i$, in contrast to the uniform grid sampling method.

Importance sampling naturally distributes the sampling points according to the weight of the PSF (a Gaussian, in this case), which implies that more samples will be taken close to the centre of the PSF where the relative weight is large. This in turn reduces the variance of the Monte Carlo integral I, which reduces the number of samples required to reach a specified level of accuracy. In addition, the inverse transform sampling method can theoretically generate points in the far tails of the Gaussian, which implies that the PSF is not artificially truncated at a certain size. This minimises the distortion of lower frequencies associated with a fixed-size uniform sampling grid.

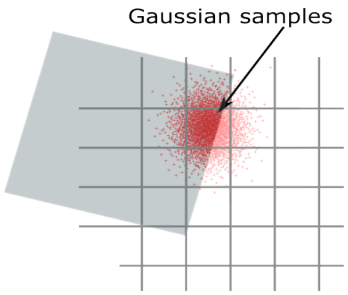

Fig. 10. Importance sampling with a Gaussian distribution

An efficient implementation of this importance sampling approach is to pre-compute the values of $\mathbf{x}_i$ using a Gaussian centered at $(0, 0)$. The sampling positions $\mathbf{x}_i$ can then translated to the pixel centred at $\mathbf{p} = (x, y)$, thereby avoiding the need to recompute sampling points for each pixel (Figure 10).

D. Gaussian PSF numerical integration

An alternative integration technique is applicable to Gaussian PSFs if an acceptably accurate approximation to the error function $\mathrm{erf}(x)$ is available. Starting from Equation 4, the polygon is partitioned into horizontal strips. In the limit, an infinitely thin strip reduces to the one-dimensional integral along the line $y = y_c$:

$$I_{y_c} = \int_{P_l(y_c)}^{P_r(y_c)} f(x)\mathrm{d}x \tag{7}$$

where $P_l(y_c)$ and $P_r(y_c)$ denote the left and right x values of the intersections of the polygon P with the line $y = y_c$. This definition only allows for convex polygons, but the extension to concave polygons will be analogous to that used to rasterise concave polygons.

The $\mathrm{erf}(x)$ function can be harnessed to derive a closed form solution to the integral in Equation 7, to yield

$$I_{y_c} = \mathrm{erf}(P_r(y_c)) - \mathrm{erf}(P_l(y_c)), \tag{8}$$

assuming that appropriate standardisation has been applied to $P_r(y_c)$ and $P_l(y_c)$.

Equation 8 provides a closed-form solution to the integral along any given horizontal slice through the polygon P. This allows us to perform numerical integration, using the adaptive version of Simpson's method, to compute the integral over all of P by integrating over the range of y values spanned by P. Figure 11 illustrates the integral that is computed for a wide Gaussian PSF centered at a pixel close to the boundary of a square target pattern. This method supports general Gaussian PSFs, including astigmatic Gaussian PSFs with full covariance matrices. If the PSF's axes are rotated with respect to the reference frame, then the simplest strategy is to rotate the target polygon to ensure that cross-sections along the integration axes are separable.

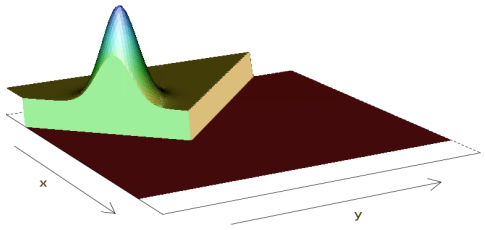

Fig. 11. Gaussian PSF bounded by target polygon. Area under curve is desired image intensity for pixel at centre of Gaussian peak

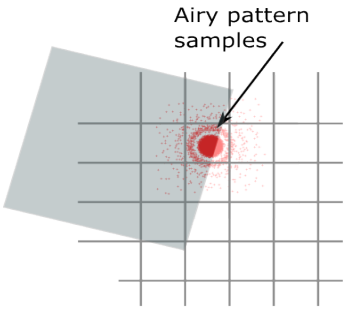

Fig. 12. Importance sampling with an Airy pattern distribution

This particular method can be exceptionally accurate, depending on the parameters of the adaptive numerical integration routine. It is possible to choose these parameters so that the computational complexity is comparable to that of the importance sampling rendering method, but yielding higher accuracy synthetic images.

E. Diffraction + box function importance sampling

Equation 3 is appropriate for rendering simple point spread functions, but does not address compound PSFs, such as the system PSF of a square photosite aperture PSF combined with a circular aperture diffraction pattern PSF. It is possible to perform the convolution of these two PSFs as a preprocessing step, thereby obtaining a single PSF which could be used in a table-driven importance sampling scheme.

A more elegant solution is to combine the area-weighted rendering strategy directly with the importance sampling scheme. Consider the set of sampling positions generated from a Gaussian distribution, as described in Section III-C. Rather than computing the Monte Carlo integral of this Gaussian PSF convolved with the target polygon indicator function, we can replace the indicator function test with a step that computes the area of the intersection of the target polygon and a square polygon (with photosite pitch side lengths) placed at each sampling position. This process thus performs the convolution of the target polygon and the photosite aperture box function first, using this result to compute the Gaussian PSF convolution using importance sampling.

To extend this method to a circular aperture diffraction PSF, we simply replace the Gaussian-distributed sampling positions with samples following the appropriate Airy pattern distribution (Figure 12). The Airy pattern distribution of samples is obtained through a look-up table that approximates the cumulative Airy pattern distribution.

F. Diffraction + OLPF importance sampling

The method described in Section III-E can be extended to render the effects of a 4-dot OLPF. Rather than computing the intersection of a single square with the target polygon at each sampling position, we instead compute the average of four such intersections, with each square placed at the appropriate offset as defined in the OLPF's specification. This approach is, of course, four times more computationally expensive.

G. Spectral sampling

Diffraction effects are wavelength dependent, which may have significant implications on computational complexity if wide-band panchromatic systems are to be simulated, since the most accurate simulation would involve rendering and blending synthetic images at multiple wavelengths, and combining them with the appropriate spectral-response weighting. Simulation of synthetic images intended for algorithms running on a Bayer Colour Filter Array (CFA) sensor (which covers most colour cameras) would require rendering at least three separate synthetic images (one for each band), possibly more if the colour filters are particularly wide. Fortunately, many algorithms (e.g., ellipse-detectors) can be verified at a single wavelength.

IV. Performance Evaluation of Rendering Strategies

A. Comparison of Gaussian PSF rendering accuracy

The following rendering algorithms were tested:

- $11 \times 11\times$UP is a uniform sampling strategy of 11×11 points centered around the target pixel. The sampling positions are truncated to the nearest integer to represent a standard linear filter without any sub-pixel sampling. This is equivalent to applying a Gaussian filter after rendering the synthetic image with one sample per pixel.
- $11\times11\times$U is a uniform sampling strategy of 11×11 points, but the sampling points are scaled relative to the desired Gaussian width. Sub-pixel spacing is used.
- $121\times$IS is an importance sampling method, with 121 (i.e., 11×11) samples from the same Gaussian distribution as that specified in the PSF. Sub-pixel spacing is used.
- $2025\times$IS is an importance sampling method, with 2025 samples drawn from the same Gaussian distribution as that specified in the PSF. Sub-pixel spacing is used.
- NI is a numerical integration implementation relying on an adaptive version of Simpson's rule (Section III-D).

These algorithms were evaluated over a range of images with Gaussian PSFs. Different standard deviation values were selected to evaluate performance over both small and large (relative to pixel size) PSFs. In addition, the sub-pixel position of the step edge was varied over 25 sub-pixel offsets to produce a more accurate assessment of algorithm performance.

The MTF50 metric is defined as the resolution at which the MTF curve reaches a contrast value of 50%, and is generally considered as a measure of resolution that correlates well with subjective human judgement of the sharpness of an image. For a Gaussian PSF, the relationship between MTF50 and standard

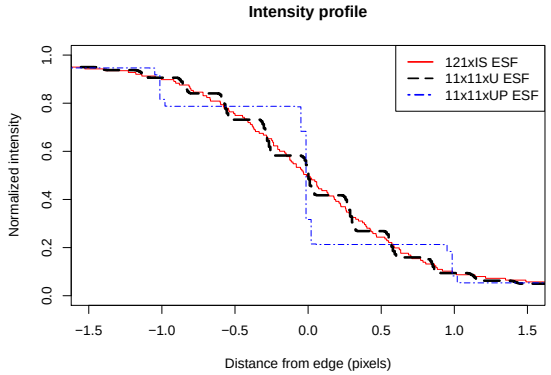

Fig. 13. Edge spread functions of 121×IS, 11 × 11×U and 11 × 11×UP rendering methods.

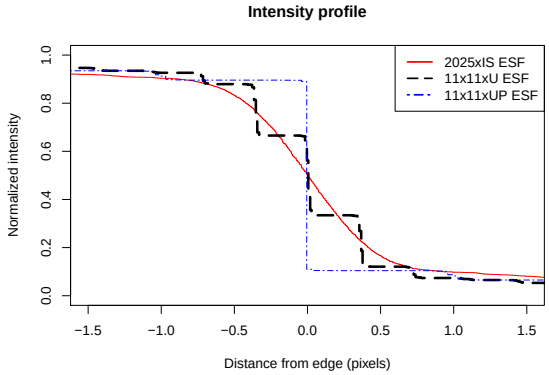

Fig. 14. Edge spread functions of 2025×IS, 11 × 11×U and 11 × 11×UP rendering methods.

deviation is fixed, hence standard deviation may be expressed as an MTF50 value in cycles per pixel. The range investigated in Table I runs from MTF50=0.1 (equivalent to Gaussian SD of 1.874) to MTF50=0.4 (equivalent to an SD of 0.468).

Figure 13 illustrates the intensity profile across the step edge subject to a Gaussian PSF with SD=0.625 (MTF50=0.3), rendered using the 11 × 11×U and 121×IS algorithms. None of the curves are smooth (compared to the expected exact Gaussian integral), but it is clear that the importance sampling algorithm is significantly closer to the desired curve (not shown). Table I confirms that the RMSE of the importance sampling algorithm is roughly 4 times smaller than the uniform grid sampling algorithm for the illustrated case, and that the integer-pixel grid uniform sampling algorithm (11 × 11×UP) fails miserably with such narrow PSFs.

The edge profile of the direct numerical integration algorithm (NI) is so accurate that it differs from the expected analytical profile only in the least significant bit of the 16-bit values used to represent intensities, i.e, differences are of the same magnitude as potential rounding errors. This rendering algorithm is therefore suitable for creating reference images.

To assess the impact of PSF accuracy on a real-world application, the slanted-edge algorithm was used to evaluate the MTF50 values of the various synthetic images. The results are shown in Table II. Even though the RMS errors of the NI method were significantly smaller than those of the 121×IS and 2025×IS algorithms, it appears that this does not translate into smaller errors in the MTF50 values as measured by the slanted-edge algorithm. One potential explanation is that the slanted edge method is more sensitive to lower spatial frequencies, so that the apparent roughness of the 121×IS algorithm (seen in Figure 13) manifests mostly at frequencies above Nyquist. The result is that additional accuracy in the PSF (as offered by the 2025×IS and NI algorithms) offers no real-world advantage for the slanted-edge algorithm.

B. Comparison of Airy pattern PSF rendering accuracy

The accuracy of the algorithms of Section IV-A were evaluated on an Airy pattern PSFs; the NI algorithm cannot be applied to the Airy pattern, and has been replaced by an importance sampling algorithm set to take 40401 samples per pixel. The simulated pixel pitch was fixed at 4.88 micron, and green light (550 nm) was chosen to compute the diffraction pattern. Different numerical apertures were investigated, since this controls the effective width of the Airy pattern relative to the pixel size.

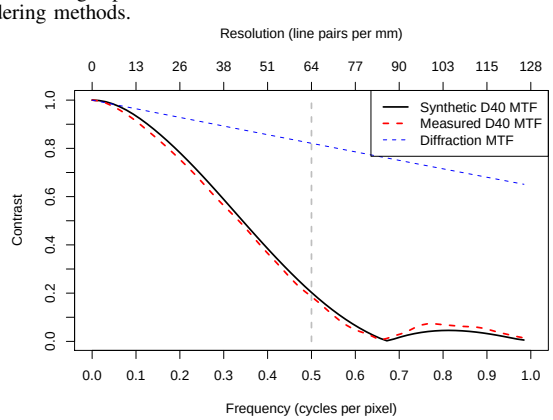

Fig. 15. Comparison of the MTF of a synthetic image with that of a knife-edge target imaged with a Nikon D40 camera.

From Table III is can be seen that the importance sampling algorithms once again have a decisive lead over the uniform sampling strategies (also visible in Figure 14). It does appear that the accuracy improves very slowly with an increase in the number of samples taken. One of the main reasons for this apparent slow increase is the large support of the Airy pattern. Since the importance sampling algorithms have been limited to a radius of 18 units (scaled according to f-number), a significant part of the tail of the Airy pattern is being truncated. This results in a lower limit on the RMSE values computed on the ESF, which cannot be reduced by increasing the number of samples while keeping this radius fixed.

C. Demonstration of system PSF accuracy

The accuracy of the combined PSF rendering strategy discussed in Section III-F is demonstrated in Figure 15. A Nikon D40 camera was used to image a knife-edge target, after which the slanted-edge algorithm was used to obtain the empirical MTF of the combined lens, OLPF and photosite aperture system. Focus bracketing was used to ensure that the system MTF is as accurate as possible, and a lens that is known to be diffraction limited was used. The MTF curve extracted

TABLE I
MEAN RMSE FOR GAUSSIAN PSFS WITH DIFFERENT STANDARD DEVIATIONS, OVER 25 DIFFERENT SUB-PIXEL SHIFTS.

Target	Mean RMS error ± standard deviation				
MTF50	11 × 11×UP	11 × 11×U	121×IS	2025×IS	NI
0.10	1.47e−02 ± 4.00e−05	1.24e−02 ± 1.68e−05	3.04e−03 ± 1.39e−05	4.05e−04 ± 2.01e−06	4.80e−06 ± 1.04e−07
0.15	1.94e−02 ± 7.91e−05	1.00e−02 ± 1.33e−05	2.44e−03 ± 1.44e−05	3.28e−04 ± 2.36e−06	3.88e−06 ± 7.87e−08
0.20	2.32e−02 ± 2.66e−05	8.62e−03 ± 2.01e−05	2.10e−03 ± 1.11e−05	2.80e−04 ± 2.86e−06	3.33e−06 ± 7.72e−08
0.25	2.70e−02 ± 6.81e−05	7.69e−03 ± 1.32e−05	1.87e−03 ± 1.21e−05	2.50e−04 ± 3.05e−06	2.94e−06 ± 8.54e−08
0.30	3.07e−02 ± 3.62e−05	7.00e−03 ± 1.72e−05	1.71e−03 ± 1.57e−05	2.28e−04 ± 2.96e−06	2.70e−06 ± 9.95e−08
0.35	3.47e−02 ± 3.12e−05	6.47e−03 ± 2.65e−05	1.58e−03 ± 1.42e−05	2.11e−04 ± 2.06e−06	2.48e−06 ± 8.16e−08
0.40	3.82e−02 ± 6.14e−05	6.02e−03 ± 1.14e−05	1.48e−03 ± 1.86e−05	1.97e−04 ± 2.77e−06	2.32e−06 ± 6.53e−08

TABLE II
MEAN MTF50 ACCURACY EVALUATION FOR GAUSSIAN PSFS WITH DIFFERENT STANDARD DEVIATIONS, OVER 25 DIFFERENT SUB-PIXEL SHIFTS.

Target	Mean error (%) ± standard deviation				
MTF50	11 × 11×UP	11 × 11×U	121×IS	2025×IS	NI
0.10	3.750 ± 0.016	5.206 ± 0.050	3.296 ± 0.036	3.343 ± 0.025	3.389 ± 0.008
0.15	1.513 ± 0.039	3.722 ± 0.059	1.563 ± 0.077	1.494 ± 0.028	1.521 ± 0.012
0.20	0.819 ± 0.060	3.039 ± 0.066	0.868 ± 0.086	0.787 ± 0.035	0.806 ± 0.014
0.25	0.743 ± 0.086	2.752 ± 0.046	0.555 ± 0.057	0.394 ± 0.030	0.414 ± 0.022
0.30	3.906 ± 0.125	2.420 ± 0.066	0.181 ± 0.177	0.119 ± 0.037	0.137 ± 0.036
0.35	24.573 ± 0.347	2.662 ± 0.080	0.066 ± 0.050	0.113 ± 0.051	0.103 ± 0.051
0.40	150.000 ± 0.000	2.063 ± 0.100	0.278 ± 0.143	0.327 ± 0.073	0.321 ± 0.075

TABLE III
MEAN RMSE FOR AIRY PATTERN PSFS AT DIFFERENT APERTURE VALUES, OVER 25 DIFFERENT SUB-PIXEL SHIFTS.

Relative	Mean RMS error ± standard deviation				
aperture	11 × 11×UP	11 × 11×U	121×IS	2025×IS	40401×IS
f/2.8	3.00e−02 ± 5.76e−05	1.00e−02 ± 4.53e−05	3.55e−03 ± 1.97e−05	1.80e−03 ± 1.43e−06	1.68e−03 ± 1.06e−06
f/5.6	3.85e−02 ± 5.61e−05	1.43e−02 ± 5.83e−05	5.02e−03 ± 1.13e−05	2.52e−03 ± 1.11e−06	2.34e−03 ± 6.48e−07
f/8	4.12e−02 ± 5.00e−05	1.71e−02 ± 5.83e−05	5.94e−03 ± 9.97e−06	2.86e−03 ± 6.12e−07	2.63e−03 ± 3.15e−07
f/16	2.86e−02 ± 2.37e−05	2.41e−02 ± 5.40e−05	7.75e−03 ± 6.24e−06	3.03e−03 ± 9.18e−07	2.84e−03 ± 9.17e−07
f/32	3.22e−02 ± 1.88e−05	3.37e−02 ± 3.63e−05	8.61e−03 ± 7.62e−06	3.15e−03 ± 3.49e−06	2.91e−03 ± 3.26e−06

from the synthetic image matches the empirical camera MTF reasonably well.

D. Rendering time

Due to space constraints, detailed rendering time results have been omitted, but brief results follow. All synthetic images were rendered as 446×446 pixel images, containing a single square target of 250×250 pixels in size. Rendering a Gaussian PSF (standard deviation of 0.625 pixels) and an Airy pattern PSF (f/8, $\lambda = 0.55\mu$m, pitch$= 4.88\mu$m) yields the following rendering times:

Gauss. Alg.:	11 × 11×U	121×IS	2025×IS	NI
Time (s) :	1.1	3.4	4.15	42.38
Airy Alg.:	11 × 11×U	121×IS	2025×IS	40401×IS
Time (s) :	0.8	0.819	3.65	67.7

Rendering times depend somewhat on the diameter of the PSF, with wider PSFs rendering more slowly, owing to an adaptive early convergence test. Including the effects of the photosite aperture is expensive: the 2025×IS rendering times increase to 29.5 s and 119 s for the single-photosite aperture and 4-dot OLPF simulations respectively.

V. CONCLUSIONS

This paper described a variety of rendering algorithms that may be applied to generate synthetic images with specific point spread functions. These algorithms have been demonstrated to be very accurate, while keeping the computational complexity relatively low. The results highlight that simple strategies (e.g., fixed-grid uniform sampling) produces much worse results than the importance sampling methods for the same number of samples.

The rendering methods introduced here can be used to generate reference synthetic images to the desired level of accuracy, and are available in the MTF Mapper project (http://sourceforge.net/projects/mtfmapper). These images can be used to calibrate other algorithms, e.g., the slanted-edge MTF estimation algorithm, or to evaluate super-resolution or shape-detection algorithms.

REFERENCES

[1] K. Kohm, "Modulation transfer function measurement method and results for the orbview-3 high resolution imaging satellite," in *Congress International Society for Photogrammetry and Remote Sensing*, vol. 20, 2004, pp. 12–23.

[2] S. Van der Walt, "Super-resolution imaging," Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2010.

[3] J. Ouellet and P. Hébert, "Precise ellipse estimation without contour point extraction," *Machine Vision and Applications*, vol. 21, no. 1, pp. 59–67, 2009.

[4] I. Sutherland and G. Hodgman, "Reentrant polygon clipping," *Communications of the ACM*, vol. 17, no. 1, pp. 32–42, 1974.

[5] G. Airy, "On the diffraction of an object-glass with circular aperture," *Transactions of the Cambridge Philosophical Society*, vol. 5, p. 283, 1835.

[6] R. Palum, "Optical antialiasing filters," in *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, R. Lukac, Ed. Boca Raton, FL: CRC Press, Sept. 2008, pp. 105–136.

[7] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical recipes: The art of scientific computing*, 3rd ed. Cambridge university press, 2007.

[8] B. Moro, "The full Monte," *Risk*, vol. 8, no. 2, pp. 57–58, 1995.