1 The derivation of the ZOGY difference image

In this document, we study some practical issues of performing the ZOGY subtraction and its relation to the Alard–Lupton (AL) method[1] combined with the decorrelation afterburner. We assume that the reader is somewhat familiar with the Zackay–Ofek–Gal-Yam (ZOGY) algorithm deduction steps in the ZOGY paper[2] Appendix A.

We recall four key ideas behind the ZOGY subtraction method: a) Given an image with uncorrelated, homoscedastic pixel noise (the noise variance in each pixel is the same value all over the image), its convolution with an arbitrary kernel leads to noise correlation between the resulting pixels. However, in frequency space, the frequencies remain independent (as random variables), only the amplitude (variance) of their noise content changes. b) The independent frequency space pixels are complex Gaussian random variables. For a signal detection purpose, similar log probability expressions can be written as for the real-valued random variables (image pixels). c) Log probability can be calculated as a sum over all the independent frequencies, weighting the squared absolute difference at each frequency with its inverse noise variance. d) The detection statistic in frequency space can be split into two multiplicative terms. In image space, these two terms can be interpreted as a difference image and its PSF, which produces a per-pixel detection statistic by convolution. The difference image is constructed in frequency space so that each frequency bin is a (complex) random variable and has the same variance. This implies that the difference image in image space has uncorrelated (and under the model assumptions), homoscedastic noise in its pixels. The difference image noise remains uncorrelated despite the PSF matching procedure. We recall the following equations from the ZOGY paper. The new N science and the reference R images are modeled as:

$$R = F_r T \otimes P_r + \epsilon_r \tag{1}$$

$$N = F_n T \otimes P_n + \epsilon_n \tag{2}$$

where F_r, F_n are photometric scaling constants, T is the *truth* image, P_r, P_n are the image PSFs and ϵ_r, ϵ_n are per-pixel Gaussian white noise with homogenous variance in the images.

The detection statistic of N having a different T value than R at any pixel position can be written in frequency space as:

$$\hat{S} = \frac{F_n F_r^2 \overline{\hat{P}_n} \left|\hat{P}_r\right|^2 \hat{N} - F_r F_n^2 \overline{\hat{P}_r} \left|\hat{P}_n\right|^2 \hat{R}}{\sigma_r^2 F_n^2 \left|\hat{P}_n\right|^2 + \sigma_n^2 F_r^2 \left|\hat{P}_r\right|^2} \tag{3}$$

In image space, S is called the score or significance image and represents the significance of a source detection for each pixel. The difference image is defined as:

$$\hat{D} = \frac{F_r \hat{P}_r \hat{N} - F_n \hat{P}_n \hat{R}}{\sqrt{\sigma_r^2 F_n^2 \left|\hat{P}_n\right|^2 + \sigma_n^2 F_r^2 \left|\hat{P}_r\right|^2}} \tag{4}$$

[1] https://ui.adsabs.harvard.edu/abs/1998ApJ...503..325A/abstract
[2] https://ui.adsabs.harvard.edu/abs/2016ApJ...830...27Z/abstract

and its PSF:

$$\hat{P}_D = \frac{F_r F_n \hat{P}_n \hat{P}_r}{F_D \sqrt{\sigma_r^2 F_n^2 \left|\hat{P}_n\right|^2 + \sigma_n^2 F_r^2 \left|\hat{P}_r\right|^2}} \tag{5}$$

so that:

$$\hat{S} = F_D \hat{D} \overline{\hat{P}_D} \tag{6}$$

The difference image (and similarly the score image) can be written as the difference of two "matched" images as in Equation (7).

$$\hat{D}_z = \frac{\frac{\hat{P}_r}{F_n}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_r\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\hat{P}_n\right|^2}} \hat{N} - \frac{\frac{\hat{P}_n}{F_r}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_r\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\hat{P}_n\right|^2}} \hat{R} = \hat{c}_n \hat{N} - \hat{c}_r \hat{R} \tag{7}$$

Here $\hat{c}_n$ and $\hat{c}_r$ are the matching kernels for the original science and template images. If the original image PSFs are accurately described by P_r, P_n, then the frequency space multiplications transform the PSFs of the two images to be identical, P_D, Equation (5). Note that while we followed the terminology of the ZOGY paper here and referred to the images as science and reference images, the entire ZOGY method is symmetrical to the swapping of the images. In the following, we may simply denote images with 1 and 2 indices.

2 Discussion points

We list the following questions that can define the direction of future ZOGY image differencing code development in the LSST stack.

How does an ideal Gaussian PSF point source look like theoretically in a ZOGY difference image? Discussed in Section 3. In Section 4, we look for answers: What causes the extensive, oscillating visual patterns in the ZOGY difference image around certain sources and image features (Section 4.2)? What shall we do with the numerical problems that appear in certain regions in frequency space and appear as pattern artifacts in image space (Section 4.4)? Shall we implement a Gaussian PSF approximator that produces the PSF frequency space representation directly? Shall we implement a Gaussian PSF width estimation to determine which input PSF is sharper so that a realistic limiting value can be used at frequencies when both PSFs (in frequency space) are below a threshold (Section 4.4)? How shall we handle division by zero scenarios in the ZOGY difference and significance image calculation (Section 4.1)?

In the Appendix, among other smaller topics, we raise the question whether we can use zero padding for calculating the score image, or shall we use model white noise padding (Section 9.7)?

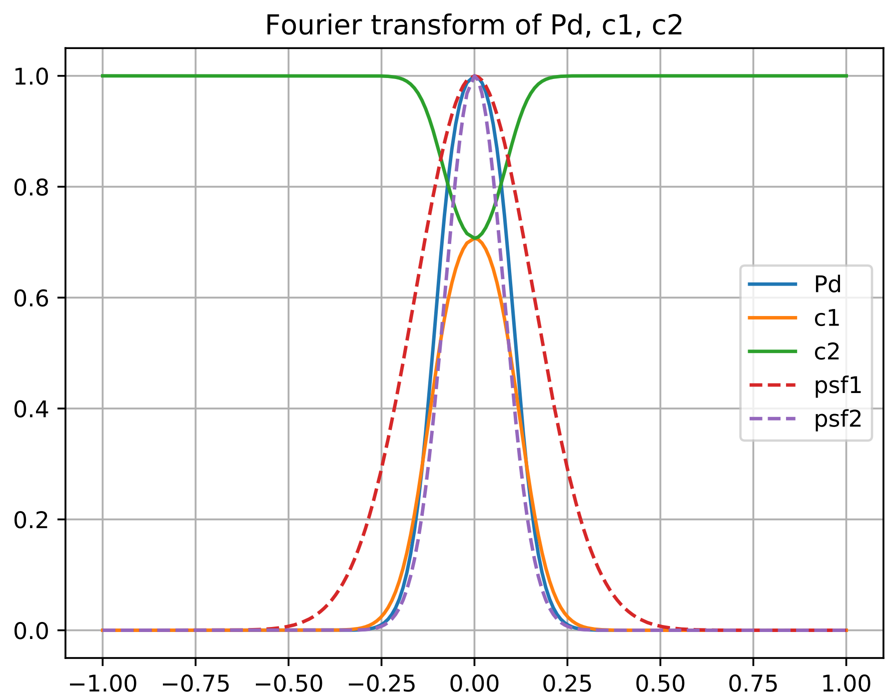

Figure 1: 1D slice along the x-axis in frequency space of the matching kernels and the PSF of the ZOGY difference image. The Fourier space representation of the input PSFs is also shown. The two PSFs have widths of $\sigma_1 = 1$, $\sigma_2 = 2$ in image space, i.e. PSF_1 is originally the narrower but the Fourier transformation swaps this relation.

3 The theoretical solution of the ZOGY matching kernel and difference image PSF

In this section, we derive numerical solutions for pure Gaussian PSFs. The inverse Fourier transforms of the ZOGY matching kernel or difference image PSF expressions are not expressible in closed symbolic forms, even in this case. We perform numerical integration of the functions.

In Figure 1, we show 1D slices of the 2D solutions of P_d, c_1 and c_2. Noise variances and photometric scaling factors are unity for simplicity. As the input PSFs are pure Gaussians, i.e. symmetric, real value functions, their Fourier transforms are also real and symmetric.[3] Note the different behavior of the two

[3] Detailed calculation notebooks are part of DM-26087.

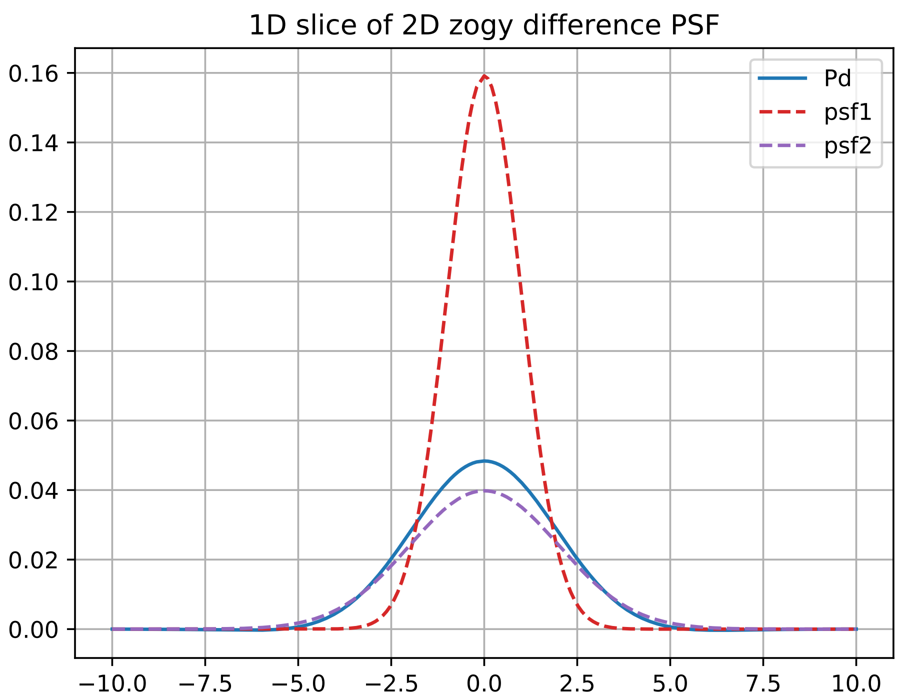

Figure 2: 1D slice along the x-axis in image space of the PSF of the ZOGY difference image with the input image PSFs.

matching kernels towards high frequencies. The matching kernel of the narrower PSF image c_1 goes to zero, while the other one goes to unity here. While the graphs of c_1 and c_2 resemble to Gaussians, they are not anymore, and we must use numerical integration to calculate their inverse Fourier transform. Their image space values for points along the x-axis is shown in Figures 3 and 4.

In Figure 2, the PSF of the ZOGY difference image is shown. It is close to the wider input PSF, but strictly it's not a Gaussian, it has a negative overshoot, about 1% of its peak value. This means that in an ideal case, signals in a ZOGY difference image are expected to have small negative rings around their positive peaks.

For the narrower PSF input image, the matched PSF is created by convolution with the matching kernel c_1, shown in Figure 3. This matching kernel is similar to usual Gaussian blurring but slightly narrower and has a negative tail itself.

For the wider image, the matching kernel c_2 is an identity Dirac delta kernel minus a Gaussian-like

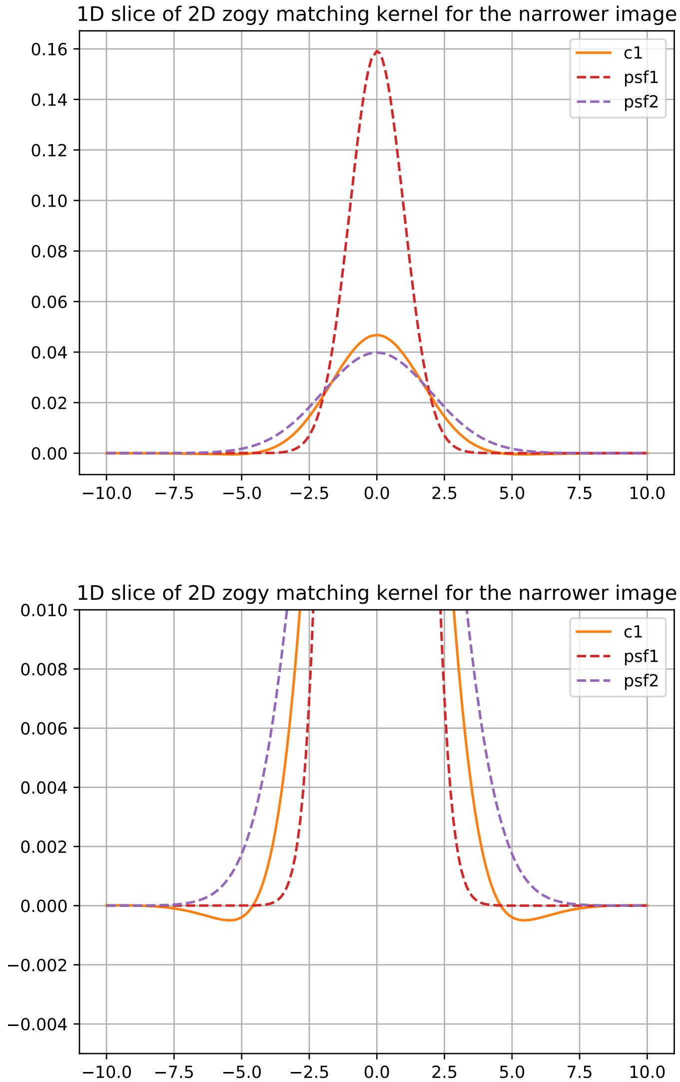

Figure 3: 1D slice along the x-axis in image space of the matching kernel for the narrower PSF image. The matching kernel is a Gaussian-like curve that has a small oscillating correction in the tails.

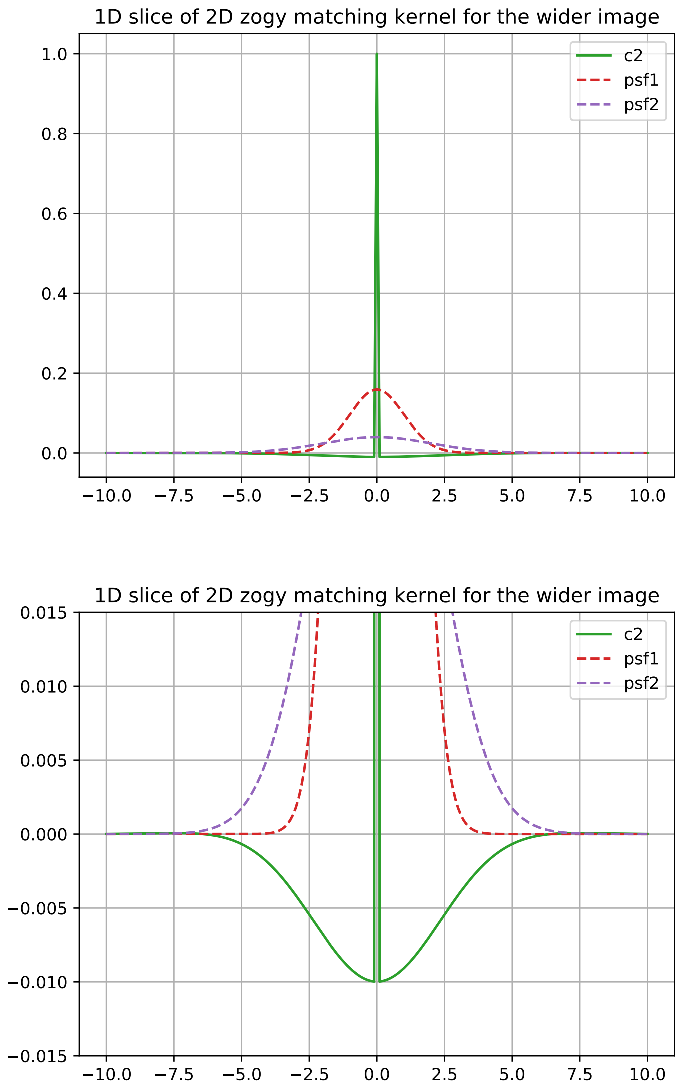

Figure 4: 1D slice along the x-axis in image space of the matching kernel for the wider PSF image. The matching kernel is the sum of a Dirac delta minus a Gaussian-like curve.

correction (Figure 4). The Dirac delta is the inverse transform of the non-zero constant level of c_2 in Figure 1 that must be subtracted for the numerical integration to converge. The Dirac delta peak is manually added to the result in Figure 4.

Note that in case of identical PSFs, both c_1 and c_2 become constant in Figure 1 which correspond to Dirac deltas in image space. I.e. the matching operation is naturally reduced to the identity operation if the two PSFs are already identical.

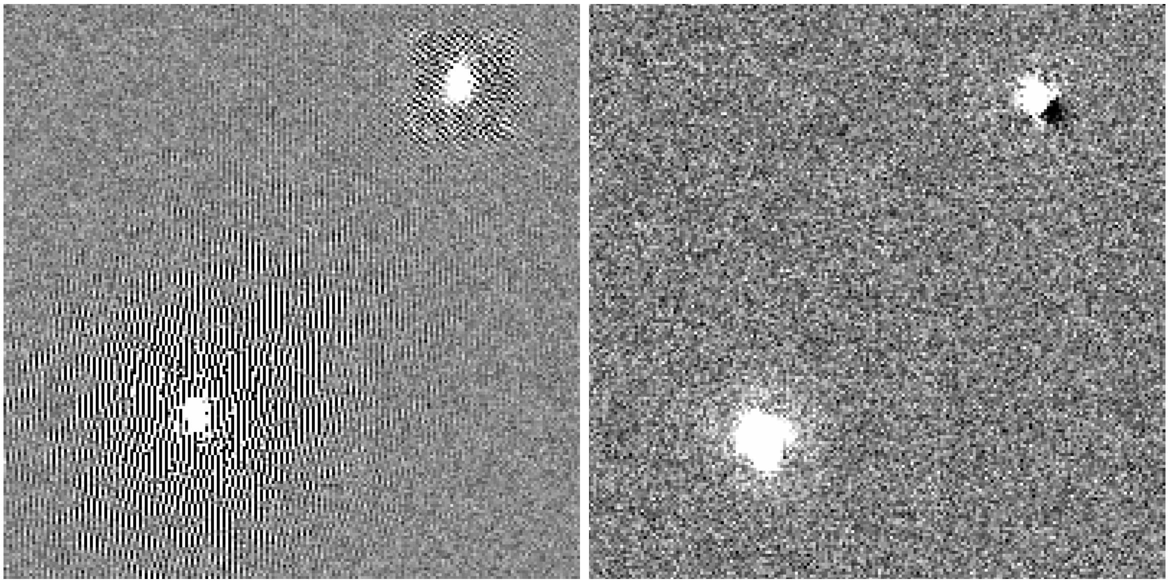

Figure 5: High frequency artifacts in the ZOGY difference image (left) around bright sources that are present in all visits in the AL processing as well. In the same visit, the AL subtraction (right) has less pronounced visual imperfections.

4 The FFT calculated matching kernel of the ZOGY difference image

In practice, the ZOGY subtraction is implemented by Fast Fourier Transforms (FFT). In this and the following sections, we study practical numerical aspects of this approach.

In Figure 5, we show typical patterns that appear around features that do not subtract well. These sources are present in all visits in the HiTS2015 data AL image difference processings and in some visits they produce different artifacts in the AL subtraction as well; however, AL artifacts are spatially more localized to the source than in the ZOGY case. The ZOGY patterns can also appear in the vicinity of masked regions, cosmic rays, or close to the image edges.

4.1 Zero values of the PSF

In Equation (4), $\hat{D}$ is not defined at frequencies where both image PSFs are zero. Indeed, according to the image models (Equation (2)), at these frequencies, the input images do not carry any information about the true image. They consist of pure noise. In accordance with this, these frequencies have zero contribution to $\hat{S}$.

We cannot allow zero division in our calculations anyway, thus we need to have a workaround for pixels where the denominator in Equations (3) and (4) are zero. We define $\hat{D}$ at these frequencies as the straightforward subtraction of the two images with the same scaling to keep the variance constant at all frequencies. Of course, $\hat{S} = 0$ at these pixels. This is currently implemented in the code stack.

4.2 Matching kernel limit values in frequency space

We saw in the theoretical solution section, that in case of Gaussian PSF-s, the matching kernels in frequency space have tails converging to different limit values. The limit values are either zero or a non-zero constant depending on whether the matching kernel belongs to the narrower or wider input PSF image, respectively.

In Figures 6 and 7, $\hat{c}_1$, $\hat{c}_2$ are calculated from two 31x31 pixel size Gaussian PSFs that were zero padded for a 1024x1024 image size, with $\sigma_1 = 3.3$, $\sigma_2 = 2.2$.[4] All numbers are real in this case. The shown frequency space images are in their natural FFT orientation with zero frequencies at the corners and highest frequencies in the centers. Starting from the corners, both solutions follow our expectations, converging either down to zero or to their expected non zero constant ($1/\sigma_{\mathrm{pixelnoise}}$) plateau. The trend breaks for both kernels in high frequency regions however, and high value noise appears.

$$\hat{c}_1 \sim \frac{1}{\sqrt{1 + \left(\frac{\hat{P}_1}{\hat{P}_2}\right)^2}} \tag{8}$$

The matching kernel limit values depend on whether $\hat{P}_1/\hat{P}_2$ is converging to zero or diverges as it can be seen in Equation (8). Once we reach the point where the Gaussian tails are dominated by noises[5], the convergence properties of these fractions become lost and the calculated matching kernel values significantly deviates from their expected limit values.

4.3 Patterns in image space

What does this mean for our calculated matching kernels back in image space?

In Figure 8, we show c_1 transformed and re-centered back to image space (but still in its fully padded image size). The purple structure indicates that there is a sign oscillation pattern all accross padded size image.

We can see that in the direction of the two axes, there are definite purplish patterns. The purple color on this red-blue color scale shows a sign oscillation that can be verified in zoomed-in versions of the figure. These patterns do not fade away in the direction of the axes from the center, indicating that these oscillating sign values have roughly the same order of magnitude absolute values. The original PSF size 31x31 cannot be clearly identified any more in the image either. We note that the appearance of these patterns is independent of the padding size. In Figure 9 P_d is shown, calculated by Equation (5). We also show the PSF of S in Figure 10. The PSF of the score image shows how a Dirac delta signal (in the truth image) appears in S, though in source detection, only the actual pixel values matter in S, the shape of the PSF does not.

[4] Detailed calculation notebooks are part of DM-26941.

[5] See Section 9.2

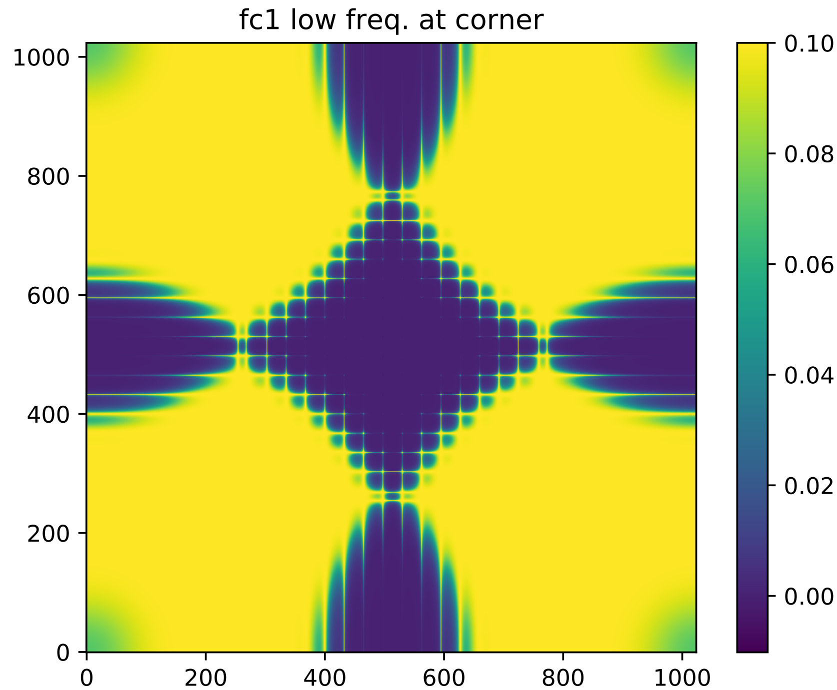

Figure 6: FFT calculated matching kernel for two Gaussian PSFs, here for the wider PSF image. This is a Fourier space image with low frequencies at the corners. In this calculation $\sigma_1 = 3.3$, $\sigma_2 = 2.2$ PSFs were generated in a 31x31 size image, that were zero padded to 1024x1024 image size before FFT. Per pixel noise variance is 100 for both images, photometric scalings are unity. All values are real due to symmetry in the inputs.

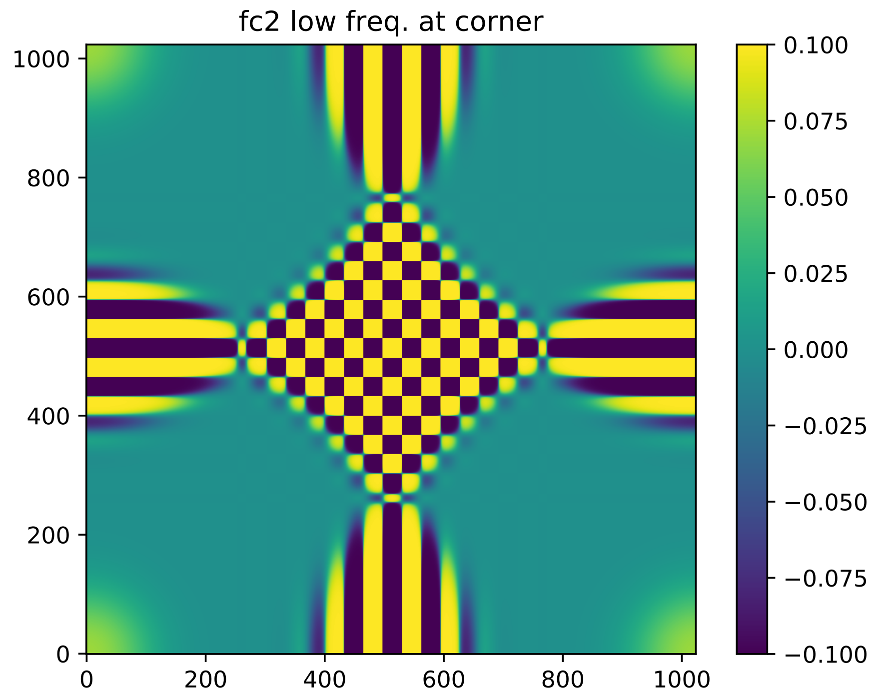

Figure 7: FFT calculated matching kernel for the narrower PSF image. This is a Fourier space image with low frequencies at the corners. All values are real due to symmetry in the inputs.

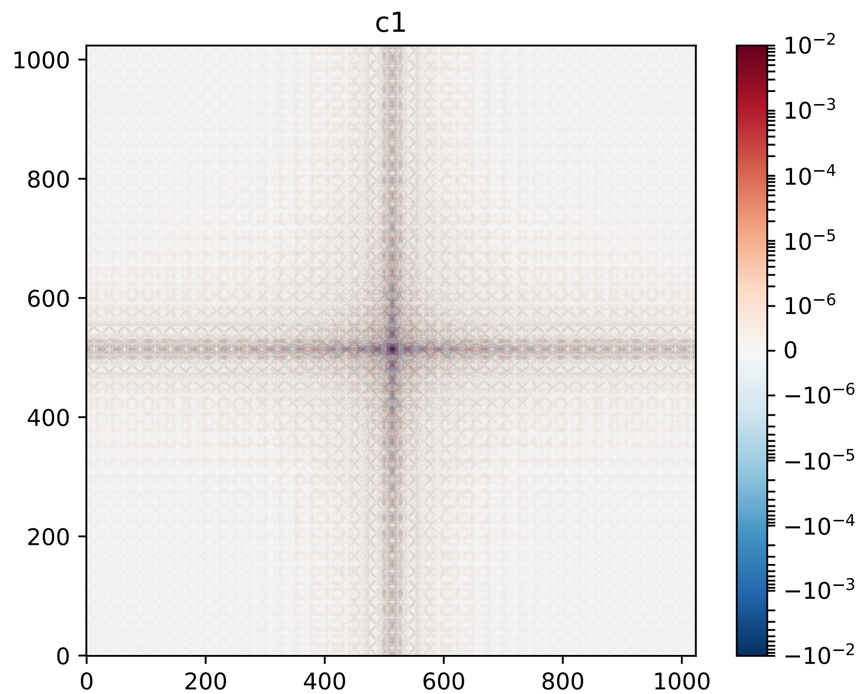

Figure 8: Two Gaussian PSFs with spatial widths of $\sigma_N = 3.3$ $\sigma_R = 2.2$ pixels. c_1, the ZOGY matching convolution in image space of the new (N) image. The purple pattern is an indication of sign oscillation all over the image.

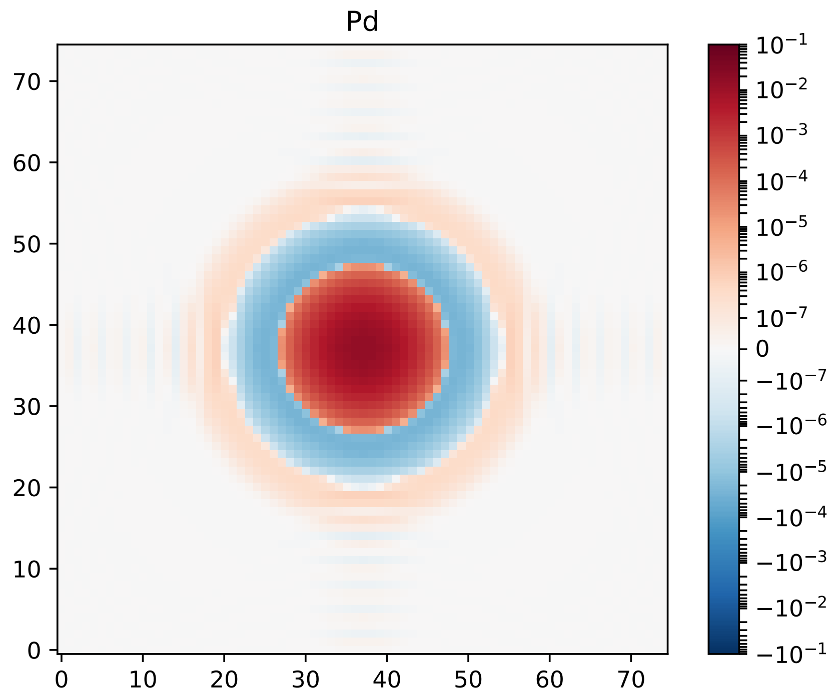

Figure 9: Two Gaussian PSFs with spatial widths of $\sigma_N = 3.3$ $\sigma_R = 2.2$. P_d, the PSF of the zogy difference image.

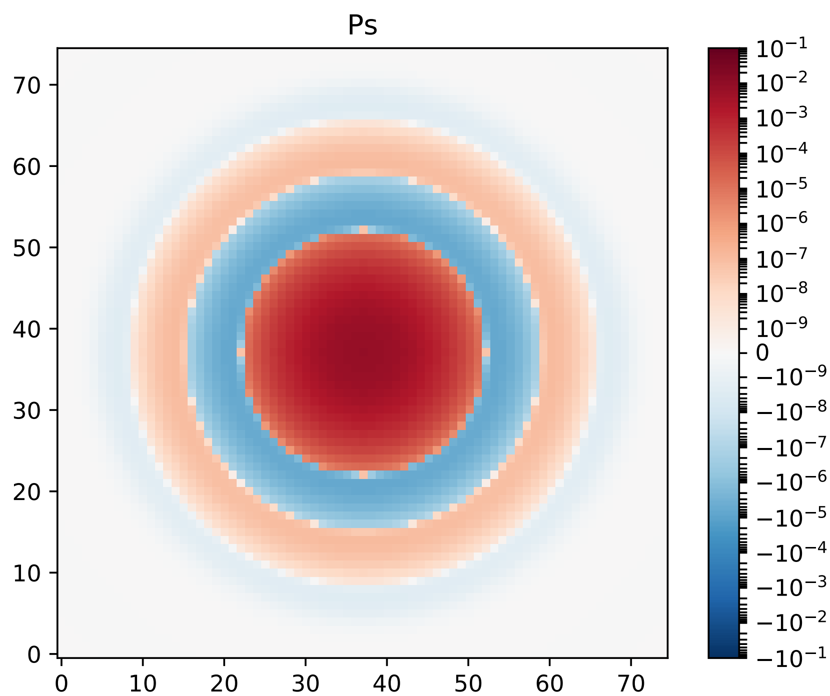

Figure 10: Two Gaussian PSFs with spatial widths of $\sigma_N = 3.3$ $\sigma_R = 2.2$ P_s, the PSF of the score image.

P_d and P_s have much cleaner images, contained in size in image space, and close to our theoretical expectations. ($\hat{P}_s \sim \hat{P}_d \overline{\hat{P}_d}$).

Recall that while we expressed P_d in Equation (5) as the function of the input PSFs, in a difference image this is the result of the convolution of the images with the matching kernels. The high frequency noise in the matching kernel is not disturbing, so long the image follows the model PSF assumption and has approximately Gaussian PSF features that suppress high frequencies. If there are edges, or signals with high frequency components in the input images, the noisy high frequency features of the matching kernel becomes visible in the difference image. Our current understanding is that the deviation of the image PSF from the model assumption and the numerical noise in the matching kernels together cause the visible artifacts in the difference images produced by the code stack. This conclusion is supported by tests on simulated images that have sources only with perfect Gaussian PSFs. In these cases, no visual artifacts can be seen.

4.4 Workaround for artifact suppression

We propose the following workarounds for the difference image artifact problem:

- In a Gaussian PSF approximation, we can directly create the PSF in the padded, full-size frequency space, avoiding the zero padding of a small image then the FFT operation. However, this approach restricts our input kernels strictly to Gaussians.

- In a more generic approach, we can still use the padded, FFT-d detected PSFs of the input images. Using a radius approximation, we can determine which input PSF is the wider one in a Gaussian approximation. Then we can introduce a configurable threshold in *frequency space* and pixels in the matching kernels can be replaced with their Gaussian limit values wherever the input PSFs go below the threshold (in absolute value, in frequency space).

- As a third option, we should recall, that the noise artifacts appear only in the difference image. In the score image, these are automatically suppressed by further convolution with P_d. We can choose to use the score image only directly for detection significance.

We repeated the above exercise by generating the Gaussian PSFs directly in frequency space and performed exactly the same matching kernel and P_d calculations. These results can be seen in Figures 11 to 13. These solutions fully satisfy the theoretical limit values and their image space counterparts are free from any noisy patterns.

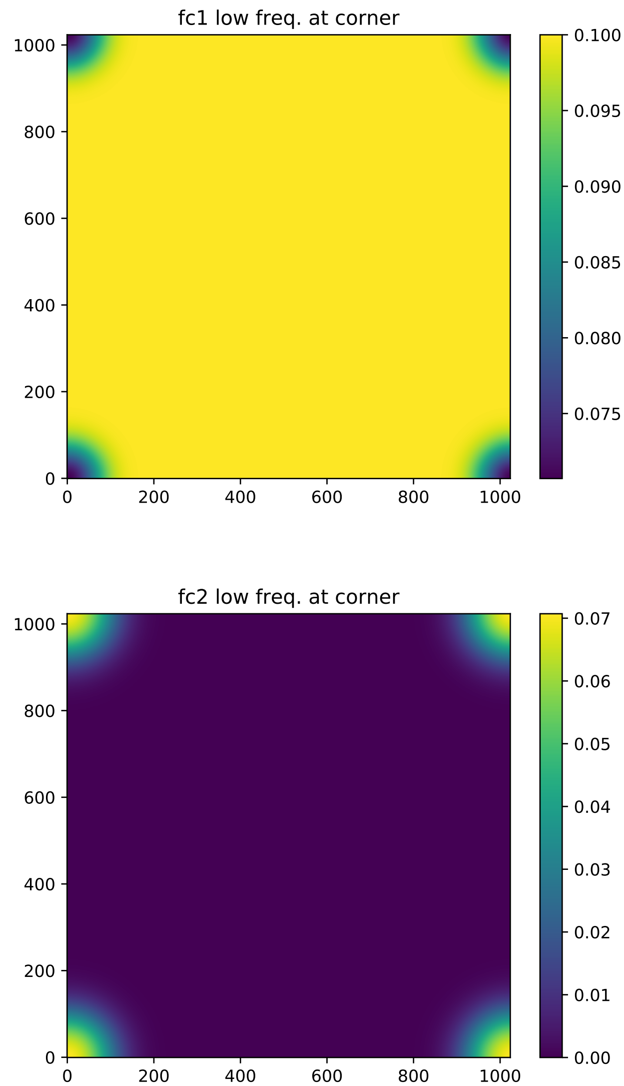

Figure 11: The matching kernels for two Gaussian PSFs in frequency space. In this calculation, PSFs were directly generated in an 1024x1024 frequency space image corresponding to image space $\sigma_1 = 3.3$, $\sigma_2 = 2.2$ widths. Per pixel noise variance is 100 for both images, photometric scalings are unity.

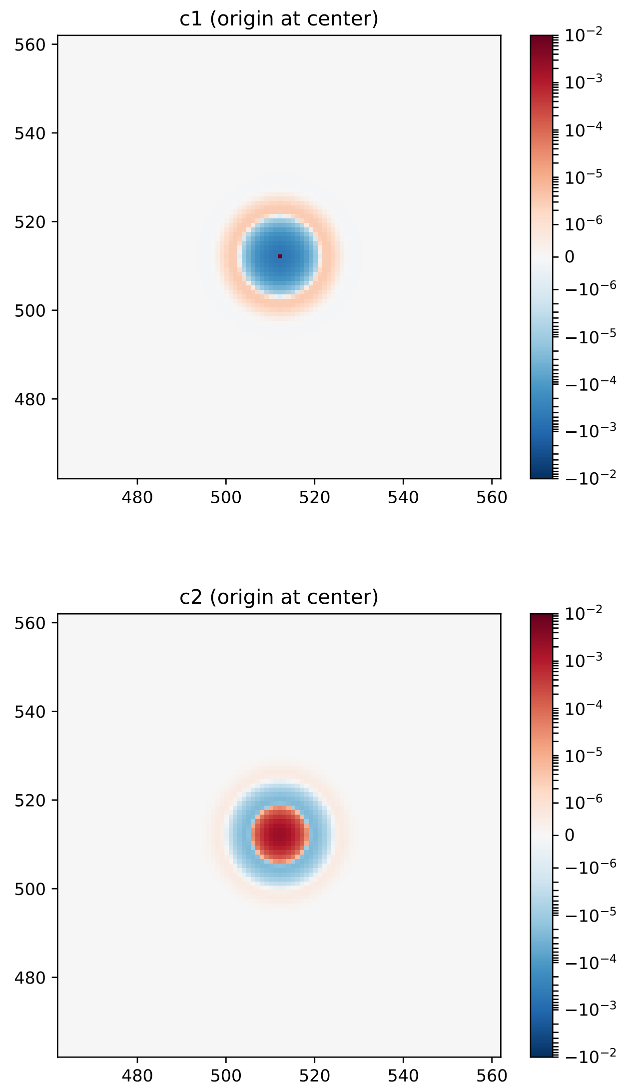

Figure 12: The matching kernels inverse FFT-d into image space, re-centered and zoomed in for details.

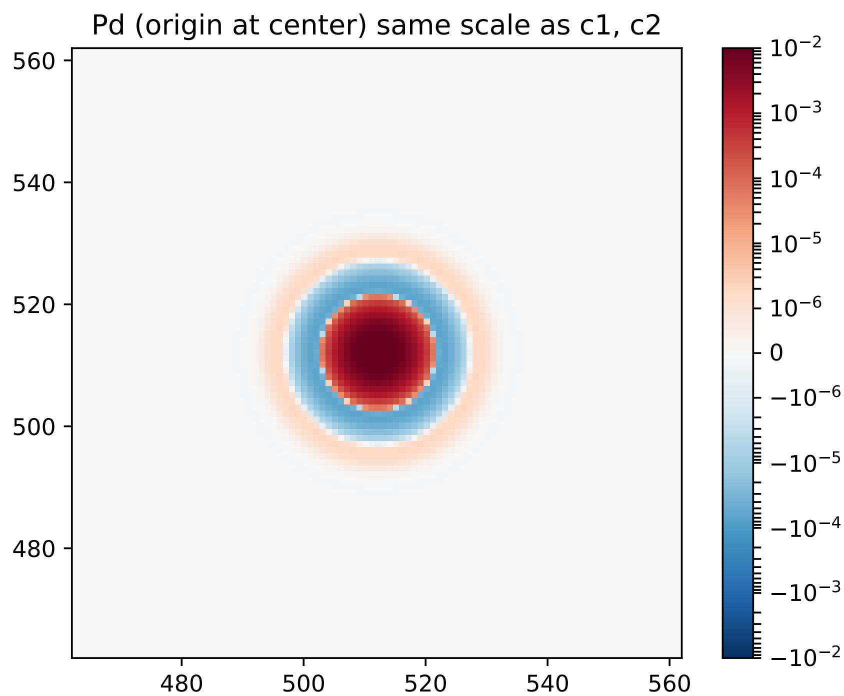

Figure 13: The difference image PSF inverse FFT-d into image space, re-centered and zoomed in for details.

5 Variance plane calculation of the difference image

While the ZOGY image model does not strictly allow for different per-pixel noise values (its noise model assumes homogeneous variance noise across all pixels), from the Equation (7) form of the difference image, we can propagate the different pixel variance information in the variance planes into the difference image. To do this, we notice that if we do convolution on an image of independent noise then, in image space, the variance plane should be convolved by the square of the convolution kernel. This is the well-known square addition of variances of independent random variables.[6]

We'd like to emphasize that this step cannot be applied to an image with already correlated noise; the square addition of pixel noise in the variance plane do not account for the covariance terms and would result in underestimation of the pixel variance. Notably, the effect of a noise decorrelation (whitening) kernel on an already convolved image cannot be applied to the variance plane based on the square addition rule as it would lower the variance further instead of reverting it to the uncorrelated level. In accordance with this, the image space square operation is not distributive with respect to convolution in general; the square of the convolution of two kernels is not the same as the convolution of the squared kernels, and we should always perform the former.

To calculate the variance plane of the difference image, we should calculate c_n, c_r in Equation (7), transform them back to image space, square them in image space, and convolve the original images' variance planes with these squared matching kernels. In practice, this convolution is more straightforward to be performed in frequency space again because these images already share common, full image size dimensions (the dimensions of our ZOGY frequency space calculations).

Equation (6) can also be written similarly to Equation (7) and the variance plane of S can be calculated this way as well.

6 ZOGY and AL equivalence

In Equation (10), we write the ZOGY score image in frequency space and expand the expression to demonstrate that the AL matching and subtraction combined with the decorrelation afterburner noise whitening theoretically leads to the same detection statistics. In the AL case the role of the two images are not symmetric, the template PSF is matched to the science image, and the science image is left intact. Assuming that the AL optimization finds the perfect matching kernel, it should be P_n/P_r. Indeed, considering the score image, the difference between the AL and ZOGY images are only a factor of $\hat{P}_r/\left|\hat{P}_r\right|$. Compared to the AL, in the ZOGY case both the difference image and its PSF carry an extra $\hat{P}_{pre}/\left|\hat{P}_{pre}\right|$ factor that cancel from the overall expression of the score image. Indeed, as $\overline{\hat{P}_r}\hat{P}_r/\left|\hat{P}_r\right|^2 = 1$, this factor can be arbitrarily added to or removed from the difference image and PSF terms without changing the resulting score image.

[6] See also ZOGY paper eqs. 26-29.

Furthermore, note that $\hat{P}_r$ from the AL perspective can be treated as an optional *pre-convolution* kernel for the science image. If we use the decorrelation afterburner with the correction term that includes the pre-convolution kernel (Equations (14) and (15)), the AL approach with the decorrelation afterburner will still lead to the same score image.

If we select the PSF of the *template* as pre-convolution kernel for the science image, the AL difference image solution should be exactly the same as the ZOGY proper difference image. However, we can use *any* pre-convolution kernel. We think that the real question is what pre-convolution kernel should be used so that the AL algorithm has a good quality stable performance in finding the matching kernel solution. The performance may vary as the matching kernel solution includes the pre-convolution kernel, and the pre-convolved science image does not have uncorrelated noise during the AL numerical optimization. Strictly speaking, correlated noise violates the optimization algorithms noise assumptions, but we don't think this is a major factor in its performance in practice. This could be studied in the future.

$$F_D = \frac{1}{\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}}} \tag{9}$$

$$\hat{S} \sim \hat{D}_Z \overline{\hat{P}_Z} = \frac{\frac{\hat{P}_r \hat{N}}{F_n} - \frac{\hat{P}_n \hat{R}}{F_r}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_r\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\hat{P}_n\right|^2}} \cdot \frac{\overline{\hat{P}_n \hat{P}_r}\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_r\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\hat{P}_n\right|^2}} = \tag{10}$$

$$= F_D \frac{\left(\frac{\hat{N}}{F_n} - \frac{\hat{R}}{F_r} \cdot \frac{\hat{P}_n}{\hat{P}_r}\right)\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}}}{\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}\left|\frac{\hat{P}_n}{\hat{P}_r}\right|^2}} \cdot \frac{\hat{P}_r}{\left|\hat{P}_r\right|} \cdot \frac{\overline{\hat{P}_n \hat{P}_r}\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_r\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\hat{P}_n\right|^2}} = \tag{11}$$

$$= F_D \hat{D}_{AL} \cdot \frac{\overline{\hat{P}_n}\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}}}{\sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}\left|\frac{\hat{P}_n}{\hat{P}_r}\right|^2}} = F_D \hat{D}_{AL} \overline{\hat{P}_{AL}} \tag{12}$$

$$\hat{D}_{AL} = \frac{\hat{D}_Z}{F_D} \cdot \frac{\left|\hat{P}_r\right|}{\hat{P}_r} \tag{13}$$

$$\hat{D}_{AL} = \frac{\frac{\hat{P}_{pre}\hat{N}}{F_n} - \frac{\hat{P}_n \hat{P}_{pre} \hat{R}}{\hat{P}_r F_r}}{\sqrt{\frac{\sigma_n^2}{F_n^2}\left|\hat{P}_{pre}\right|^2 + \frac{\sigma_r^2}{F_r^2}\left|\frac{\hat{P}_n \hat{P}_{pre}}{\hat{P}_r}\right|^2}} \sqrt{\frac{\sigma_n^2}{F_n^2} + \frac{\sigma_r^2}{F_r^2}} \tag{14}$$

7 Decorrelation afterburner normalization

To preserve the photometric scale of the AL difference image (that is the same of the science image), the decorrelation afterburner correction should be overall a convolution with a normalized correction kernel (in image space). PSF normalization can be easily ensured in frequency space. If we evaluate any PSF at 0, based on Equation (28) they should give 1. The AL matching kernel is not constrained this way in general. The sum of the AL matching kernel and the optional pre-convolution kernel can be written as separate factors in the decorrelation afterburner. The correctly normalized decorrelation afterburner is written in Equation (15).[7]

$$\hat{K} = \frac{\sqrt{\frac{\sigma_n^2}{F_n^2} S_{pre}^2 + \frac{\sigma_r^2}{F_r^2} S_{mk}^2}}{\sqrt{\frac{\sigma_n^2}{F_n^2} S_{pre}^2 \left|\hat{P}_{pre}\right|^2 + \frac{\sigma_r^2}{F_r^2} S_{mk}^2 \left|\hat{P}_{mk}\right|^2}} \tag{15}$$

8 Conclusions

We studied the ZOGY difference image matching kernels for Gaussian input PSFs in this document. In the thoretical calculations (Section 3), we showed that the matching kernels have different convergence values in their tails depending whether they belong to the narrower of wider PSF input image. In practice, using FFT, these convergence properties are not well reproduced and the resulting image space matching kernels have oscillating patterns all over the image (Section 4). We concluded that this noise is still acceptable if the input images follow their PSF models and suppress high frequencies, however noise patterns appear in the difference image if the PSFs deviate. This noise is extended, visually unappealing, and can disrupt other algorithms' performance on the difference image; however, it has little impact on the source detection statistic.

We tested the direct Gaussian PSF generation in frequency space as a possible way to avoid the convergence problems in our calculations. We expect that it would produce difference images without large scale patterns for all inputs. It is a strong restriction on the PSFs, so we also plan to look for weaker constraints in suppressing the artifacts in the difference image. We also need to consider sampling (aliasing) details before implementation.

In Section 5 we discussed how to properly calculate the variance plane in frequency space when we have noise whitening decorrelation operations.

In Section 6, we demonstrated that the AL method combined with the decorrelation afterburner leads to the same detection statistic as the ZOGY method. With preconvolution, they can theoretically lead to the very same difference image. We believe though that his approach would meet similar practical problems as the ZOGY subtraction has. This is a possible future topic to study.

In Section 9 we discuss various considerations that have relevance in the actual and for future frequency space image differencing code implementations.

Finally, the noisy matching kernels cause complications in implementing solutions in frequency space for spatial variations of the PSF in a large image. This is also a topic we plan to study in the future.

[7] As of writing this is not implemented in the LSST stack correctly. It is DM-25050.

9 Appendix

9.1 Notations

We follow the ZOGY paper symbol notations. Frequency space quantities are marked with ˆ, complex conjugation is marked by $\overline{x}$. Pixels of images are referred as functions (Equation (18)). Expectation value of random variables are marked by $\langle\ \rangle$.

We use the terms *image space* and *Fourier- or frequency space* to refer to the discrete Fourier transform of images. *Pixels* may refer to either space depending on the context.

$$x = \{x(0), x(1), ..., x(n)\},\ x(n) \in \mathbb{R} \tag{16}$$
$$\hat{x} = \{\hat{x}(0), \hat{x}(1), ..., \hat{x}(k)\},\ \hat{x}(k) \in \mathbb{C} \tag{17}$$
$$\tag{18}$$

9.2 Floating point values

The machine *epsilon* is the smallest positive floating point value where $1 + \varepsilon \neq 1$. This is $\approx 1e - 16$ for double precision.

The machine *tiny* is the smallest positive floating point value where the significand does not start with leading zeroes but the exponent is the smallest representable. Going below this value the floating point number looses significant digits and eventually rounds to exact zero. About $\mathrm{epsilon} \cdot \mathrm{tiny} = 0$.

Underflow to zero occurs around the order of the floating-point *tiny* value, we found, however, that this never practically happens. In all our practical PSF transformation cases FFT values cannot go a few orders below the floating-point *epsilon* that is several orders higher than the *tiny* limit (see Appendix for details). This is understandable if we consider that every pixel is a result of addition operations, where the number of terms roughly equals to the number of pixels in the image. As the PSFs are normalized, the zero frequency value is always 1, which approximately sets the exponent of these floating point calculations.

Furthermore, we usually zero pad a small PSF image to a larger image size that creates a window function effect in the padded image. The transformed image, therefore, have long oscillating tails in frequency space and we found that all pixel (absolute) values remain a few orders even above the epsilon threshold.

9.3 Complex random variables

$$\langle Z \rangle = \langle \mathrm{Re}(Z) \rangle + i \langle \mathrm{Im}(Z) \rangle \tag{19}$$
$$\langle \bar{Z} \rangle = \overline{\langle Z \rangle} \tag{20}$$

The variance and covariance of a complex random variable are defined as:

$$\mathrm{Var}(Z) \in \mathbb{R} \equiv \langle |Z - \langle Z \rangle|^2 \rangle = \langle |Z|^2 \rangle - |\langle Z \rangle|^2 \tag{21}$$

$$\mathrm{Cov}(X,Y) \equiv \left\langle (X - \langle X\rangle)\,\overline{(Y - \langle Y\rangle)} \right\rangle = \langle X\bar{Y}\rangle - \langle X\rangle\overline{\langle Y\rangle} \tag{22}$$

$$\mathrm{Cov}(X,X) = \mathrm{Var}(X) \tag{23}$$

9.4 Discrete Fourier transformation normalization convention

There is a freedom how normalization factors are placed in the forward and inverse Fourier transforms. This scales the individual values of frequency components compared to corresponding pixel space values. Usually, we do not need to worry about these scalings as the forward and inverse operation factors cancel out. However, certain frequency space relations change in their form if the normalization convention changes, most importantly for us, the expression of the convolution theorem changes. The definition of DFT usually has the following normalization convention:

$$\hat{X}(k) = \mathcal{F}[x](k) \equiv \sum_n x(n) e^{-i\frac{2\pi}{N}k\cdot n} \tag{24}$$

$$x(n) = \mathcal{F}^{-1}[\hat{x}](n) \equiv \frac{1}{N}\sum_k \hat{x}(k) e^{i\frac{2\pi}{N}n\cdot k} \tag{25}$$

In this convention, the convolution theorem (and its dual) looks like:

$$\mathcal{F}[x \otimes y] = \hat{x}\cdot\hat{y} \tag{26}$$

$$\mathcal{F}[x\cdot y] = \frac{1}{N}\hat{x}\otimes\hat{y} \tag{27}$$

Also:

$$\mathcal{F}[x](0) = \sum_n x(n) \tag{28}$$

These relations change with factors of $\sqrt{N}$ if the transform normalization changes. We must be sure that the correct convention is used by numpy. This is the default as of v1.18.

9.5 Noise variance properties in frequency space

Let's take a look at the covariance of the Fourier transform of zero expectation value pixels The complex covariance can be written as:

$$\begin{gathered}\left\langle \hat{x}(k)\overline{\hat{x}(j)} \right\rangle = \left\langle \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi}{N}kn} \sum_{l=0}^{N-1} \overline{x(l)} e^{i\frac{2\pi}{N}jl} \right\rangle = \\ \sum_{n,l=0}^{N-1} \left\langle x(n)\overline{x(l)} \right\rangle e^{-i\frac{2\pi}{N}(kn-jl)} = \sum_{n=0}^{N-1} \sigma(n)^2 e^{-i\frac{2\pi}{N}(k-j)}\end{gathered} \tag{29}$$

If $k = j$, we get the variance at each frequency. From the last expression in Equation (29), we can see that the variance is the same at all frequency and it is the sum of the individual pixel variances. Considering the normalization in the forward and inverse Fourier transformation, we can think of this as the average of the individual pixel variances, too.

This implies that using the average value of the variance plane as the variance in frequency space is actually not an approximation but the exact value.

If $k \neq j$, but the individual pixel variances are equal, then the phase factors in Equation (29) average out and we get that the covariance in frequency space is zero between different frequencies. As a similar expression and argument can be written for the pseudo-covariance, we receive that any two different frequencies are uncorrelated. This is the well-known relation that the Fourier transform of white noise is white noise. If σ_n-s are not equal however, the phase factors won't average to zero. Spatial variations of pixel noise introduce correlation in frequency space noise. The correlation in frequency space encodes the spatial distribution of σ_n values in image space.

We note that this is the case if we add zero padding to the image, because the zero padding can be seen as pixels with zero sigma noise. Also, if we change the correlation between frequencies by multiplying with frequency-dependent factors, this implies a spatial change of noise in image space, following the convolution theorem.

Finally, let's consider a white noise image that got convolved by a kernel image. From the convolution theorem, we get that in frequency space the variance becomes frequency-dependent, but different frequencies remain still uncorrelated.

We summarize these noise transformation properties in Table 1, noting the duality of variances values and correlation between pixels in image and frequency spaces. Our understanding is that correlated noise in image space can be decorrelated by scaling in frequency space so that all components have the same variance. This is one of the key ideas in the ZOGY difference image construction, that one square root of the likelihood variance weight can be assigned to the proper difference image, so that its noise gets whitened (decorrelated). (The other square root is part of the difference image PSF.)

The change of the spatial distribution of pixel sigmas follows the overall convolution (like c_n, c_r) of the original uncorrelated images. If furthermore, per pixel variances are uniform across the image, then the whitening restores uncorrelated white noise across the image.

image space	frequency space
white noise	white noise
different variance values in uncorrelated pixels	same average variance at all frequencies but correlation in noise between different frequencies
same variance but correlated pixel noise due to convolution operation	different variances at frequencies but noise between frequencies are still uncorrelated

Table 1: Summary of image space and frequency space noise properties.

9.6 The resolution of DFT space

Finite DFT transforms N pixel into N pixel in frequency space. The covered frequency range always goes from -1/2 through zero to $\frac{1}{2}\frac{1}{\text{px}}$ frequencies but the resolution depends on the number of input pixels (Figure 14). As conservation of information, the N resulting frequencies can distinguish exactly N spatial positions. The same concept is described by the interpretation that finite DFT always sees the input as if it were periodic, giving the same result as if the input were repeating in every N pixels. This also means that when we make a frequency space manipulation we must see not only the input image or kernel but the results as well to be periodic back in image space.[8]

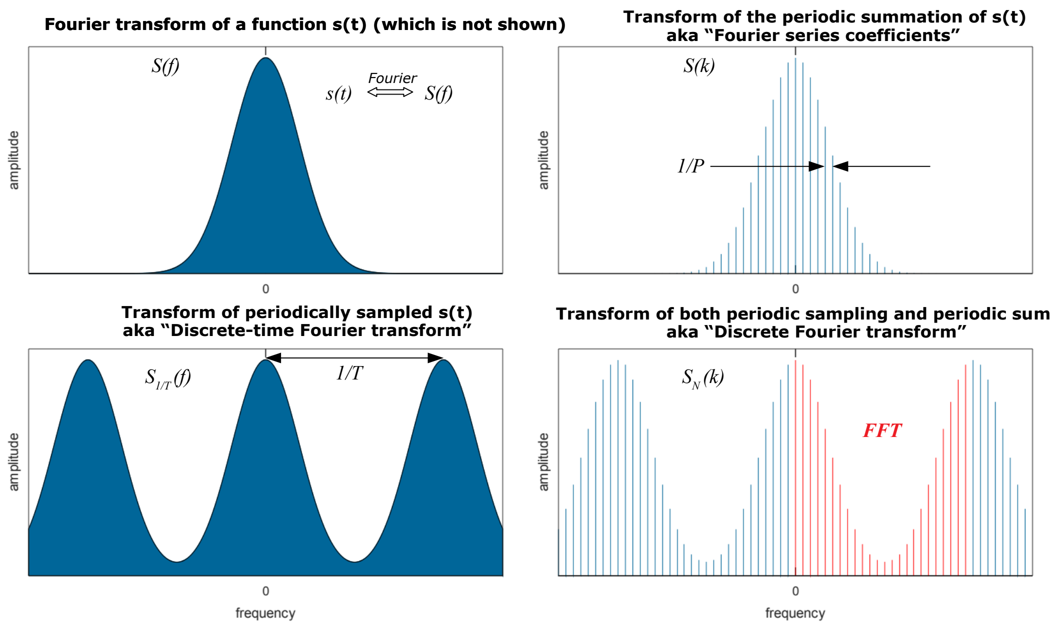

Figure 14: Overview of sampling and periodicity effects in frequency space. Given the Fourier transform of a function (top left), sampling it every T time may cause a change in the frequency space values according to the sampling theorem (bottom left). This is called aliasing, in the bottom left panel, the minimum value shown is different from the top left panel. If the function is periodic, the frequency space values reduced to discrete values as well (top right). DFT/FFT combines the two concepts (bottom right). Considering unit pixel size, the FFT space always goes to 1/2 frequency with a resolution of 1/N. Figure source: Wikipedia:Discrete Fourier transform

[8]Figure 14 source: `https://en.wikipedia.org/wiki/File:Fourier_transform,_Fourier_series,_DTFT,_DFT.svg`

9.7 Zero padding in FFT frequency space

While in image space convolution operations can have their own way of handling edges, in Fourier space, multiplication always corresponds to the circular boundary conditions in image space. If we want to implement a convolution without circular boundary conditions that we want to calculate in frequency space, we need to pad the images by extra edge pixels to avoid the reappearance of values from the opposite side. As we saw in Section 4.3, numerical artifacts in the matching kernels cannot be bounded well in image space, they fill the full area independently of the padding size. Therefore we cannot practically perform the kernel matching convolutions in image space.

In the previous section, we also saw that a zero-padding violates one of the ZOGY assumptions: that frequencies are independent and log likelihoods can be calculated from them by simple addition. Is this a significant inaccuracy in the score image?

Let's assume for a moment that the image background is extended in a sourceless way with white noise. In this case, all the assumptions of the detection statistics derivation hold thus we get Equation (6). This is a usual convolution expression in image space and at any pixel its value depends only from the half P_d size neighboring area. If P_d significant values are located in about the same square size as the original PSF size then the affected edge area also remains the same. If the PSF contains edges, however, P_d can be significantly bigger in size. Zero padding adds pixels to an image that, from a noise model perspective, all have a noise variance of zero. By padding the input images with zeroes, the pixel variance of the difference image and, in a smaller edge region, the score image variance will decrease. It is unclear whether scaling the score image S with its variance plane satisfactorily corrects for this effect. Nevertheless, this correction term is listed as a suggested rescaling of the score image in the ZOGY paper Section 3.3. Beside this correctional approach, we propose the implementation of padding with the model white noise instead of constant zeroes in the future.

9.8 Sampling

It can be shown that Gaussians with $0.95 < \sigma$, are well sampled in the sense that 3σ of their Fourier transform Gaussian fit up to the 1/2 frequency limit. (For "5σ fit" this is $1.59 < \sigma$)