



Image Subtraction in Fourier Space

Lei Hu^{1,2} , Lifan Wang³ , Xingzhuo Chen³ , and Jiawen Yang³ ¹ Purple Mountain Observatory, Nanjing 210023, People's Republic of China; lifan@tamu.edu² School of Astronomy and Space Sciences University of Science and Technology of China, 230026 Hefei, People's Republic of China³ George P. and Cynthia Woods Mitchell Institute for Fundamental Physics & Astronomy, Texas A&M University, Department of Physics and Astronomy, 4242 TAMU, College Station, TX 77843, USA

Received 2021 September 20; revised 2022 May 17; accepted 2022 May 24; published 2022 September 9

Abstract

Image subtraction is essential for transient detection in time-domain astronomy. The point-spread function (PSF), photometric scaling, and sky background generally vary with time and across the field of view for imaging data taken with ground-based optical telescopes. Image subtraction algorithms need to match these variations for the detection of flux variability. An algorithm that can be fully parallelized is highly desirable for future time-domain surveys. Here we introduce the saccadic fast Fourier transform (SFFT) algorithm we developed for image differencing. SFFT uses a δ -function basis for kernel decomposition, and the image subtraction is performed in Fourier space. This brings about a remarkable improvement in computational performance of about an order of magnitude compared to other published image subtraction codes. SFFT can accommodate the spatial variations in wide-field imaging data, including PSF, photometric scaling, and sky background. However, the flexibility of the δ -function basis may also make it more prone to overfitting. The algorithm has been tested extensively on real astronomical data taken by a variety of telescopes. Moreover, the SFFT code allows for the spatial variations of the PSF and sky background to be fitted by spline functions.

Unified Astronomy Thesaurus concepts: [Astronomy software \(1855\)](#); [Transient detection \(1957\)](#)

1. Introduction

Since Zwicky (1964), variable source identification from astronomical observations has played an indispensable role in time-domain astrophysics. However, the time and spatial variations of the point-spread function (PSF), the noises in the sources and the sky backgrounds, the optical distortions of the observing facilities, and many other effects barricade rapid and robust transient detection over wide sky areas. An efficient algorithm for calculating image differences is crucial in a broad range of astronomical observations of today. Examples include transient searches, as well as microlensing (Mao & Paczynski 1991) and pixel lensing (Crofts 1992; Baillon et al. 1993). The techniques of image differencing and co-addition are also of central importance to the next-generation survey telescope, such as the Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) and the Nancy Grace Roman Space Telescope.

To solve the PSF discrepancies between a pair of images taken at different epochs of the same field, Phillips & Davis (1995) and Tomaney & Crofts (1996) introduced a convolution kernel to match PSFs from one image to the other. They used a simple deconvolution solution to determine the kernel by calculating the ratio of the Fourier transform of a bright star or a precomputed PSF of each image. However, this approach cannot guarantee optimal subtractions since the division in Fourier space is prone to numerical instability, especially in the noise-dominated high frequencies domain (Alard & Lupton 1998; Zackay et al. 2016). Given that the PSFs are only modeled from the isolated stars with sufficient signal-to-noise ratios, the method has difficulty in using all statistically valid

pixels in kernel determination, which makes it challenging to handle very crowded fields as found in microlensing surveys. With the same goal of finding the matching kernel, Kochanski et al. (1996) moved in the first step toward kernel determination without any direct knowledge of PSFs. They solved the problem by a least-squares fitting on the image pair under consideration instead of extracting the light profiles. Nevertheless, the required computing time of the nonlinear fitting introduced by Kochanski et al. (1996) is generally formidable (Alard & Lupton 1998).

Soon after Kochanski et al. (1996), a *fork in the road* was signposted by the pioneering paper of Alard & Lupton (1998). They decomposed the convolution kernel into a basis of functions, converting the problem to be a linear least-squares question. To accommodate the varying PSFs across the image, Alard & Lupton (1998) modeled a spatially variant convolution kernel that evolves with image coordinates in a polynomial form. Subsequently, Alard (2000) simplified the calculations for constructing the least-squares matrix, making it possible to fit the kernel spatial variation with a reasonable computing cost. In the last two decades, the framework outlined in Alard & Lupton (1998) and Alard (2000) has served as the mathematical foundation upon which several successful transient survey programs have been built (e.g., Cao et al. 2016; Morganson et al. 2018; Price & Magnier 2019). As the approach does not require the existence of isolated stars, it has been applied to very crowded fields, such as microlensing observations toward the Galactic bulge (e.g., Sumi et al. 2003, 2006, 2013).

Alard & Lupton (1998) provided the essential building blocks of image subtraction, and subsequent studies continue to move the field forward, with a focus on improving the kernel basis functions and developing kernel regularization techniques. The Gaussian basis functions adopted in Alard & Lupton (1998) only allow for incomplete expansion of the convolution



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

kernels. As a result, the method is constrained by the bases' intrinsic symmetry and hinders the ability to construct a *shape-free* kernel (Becker et al. 2012). Moreover, one has to configure a variety of hyper-parameters for the kernel basis, e.g., the number and width of the Gaussians. In order to achieve sheer kernel flexibility and minimize user-adjustable parameters, Bramich (2008) and Miller et al. (2008) introduced the delta basis functions (DBFs) for the construction of complete kernel space. With the DBFs, image subtraction is capable of compensating for the sub-pixel astrometric misalignment through an unconstrained flux redistribution within the scale of kernel size (Bramich 2008; Becker et al. 2012). Albrow et al. (2009) confirmed the compelling advantage of DBFs by showing a distinct improvement in photometry accuracy over the traditional Gaussian basis functions. Bramich et al. (2013) further developed the DBF-based approach by taking the spatially varying photometric scale into account, aimed to accommodate the variation of transparency and airmass across the field in wide-field surveys.

The flip side of the ultimate flexibility of DBFs is that the resulting kernel solution can be more susceptible to noise. To alleviate the overfitting problem, Becker et al. (2012) conceived a Tikhonov regularization by adding a penalty term so that the solution would favor compact and smooth kernels. Masci et al. (2017) adopted an easily implemented way to regularize the kernel with DBFs for iPTF transient discovery. They used to solve the least-squares but dropped the eigenvalues with low statistical significance. Regularizing the kernel itself is not a unique option in resolving the overfitting problem. Bramich et al. (2016) found that using a parsimonious choice of un-regularized DBFs might be a better alternative.

Apart from the methods that stem from Alard & Lupton (1998), a numerically stable approach characterized by cross convolution was proposed by Gal-Yam et al. (2008) and Yuan & Akerlof (2008). This strategy became more prevalent since Zackay et al. (2016, hereafter **ZOGY**) presented a closed-form algorithm that yields the optimal subtraction for transient source detection. By its design, **ZOGY** can result in difference images with uncorrelated background noise. With the noise propagation during the image subtraction process, **ZOGY** claims optimal detection rates of transients with minimal subtraction artifacts.

In addition to the subtraction algorithms mentioned above, machine-learning approaches are making their way into the field. Sedaghat & Mahabal (2018) recently suggested the use of deep convolutional neural networks instead of PSF matching for transient detection. Hitchcock et al. (2021, hereafter **PyTorchDIA**) proposed a novel machine-learning approach to optimize the convolution kernel by making use of automatic differentiation in PyTorch instead of constructing a least-squares matrix.

We show in this paper that the entire imaging differencing process can be performed with fast Fourier transform (FFT). FFT can be easily adapted to different computer environments for parallel processing. Our method, saccadic fast Fourier transform (SFFT), provides a massively parallelizable framework for image subtraction and brings substantial processing speed acceleration when implemented on graphics processing units (GPUs). Besides the gain in computing cost, SFFT can retain the crucial features that have come from the advances in image subtraction since Alard & Lupton (1998), including that (1) SFFT employs flexible DBFs as kernel basis functions; (2)

SFFT can adequately handle the spatial variations of convolution kernel, photometric scale, and differential background; (3) SFFT does not rely on the availability and distribution of the observed stellar objects.

This paper is structured as follows: Section 2 provides the mathematical frameworks of SFFT. Section 3 introduces the implementation of SFFT and the details of the software pipeline. Section 4 shows the application of the SFFT pipeline to data taken from a variety of telescopes. Section 5 presents the computing performance of SFFT, with comparisons to other existing image subtraction implementations. Section 6 discusses the limitations of SFFT and plans for future works. All the code about SFFT subtraction is available on Github.⁴

2. Methodology of SFFT

2.1. Overview

As the core engine of transient detection, image subtraction is often the most computationally intensive individual task in the data processing pipeline. The ongoing trend of very massive data flow from time-domain surveys is making real-time data reduction increasingly challenging. This has motivated us to develop a new tool to relieve the computational bottleneck while reconciling the subtraction performance and the computing cost.

The image subtraction algorithms emanated from Alard & Lupton (1998) have been broadly applied in astronomical transient surveys. Our proposed method, SFFT, is a new member of this category. The main purpose of the approach aims to perform PSF matching via image convolution with pixelized kernels formulated as linear combinations of a predefined set of basis functions. The fact that astronomical images from wide-field surveys generally possess nonconstant PSF across the entire field of view leads to spatial variations of the convolution kernels. Moreover, the spatially varying photometric scale also needs to be taken into account in order to match the images from wide-field surveys.

There have been some attempts to accelerate the calculations involved in image subtraction, either aimed at faster kernel determination or speed-up of the subtraction afterward. For image subtraction, constructing the least-squares matrix to solve the convolution kernels is computationally expensive, especially for spatially variant kernels. Alard (2000) proposed fitting the space-varying kernel on a set of subareas of the field, with the simple assumption that the spatial variation of the kernel within each subarea can be negligible. The useful strategy has been adopted in the software **HOTPANTS**, a widely used implementation of Alard & Lupton (1998). Apart from the efficient simplification in computing, the flexibility of using subareas has some additional benefits in practice. One may preselect an optimal set of subareas to exclude the sources in observational data that cannot be perfectly modeled by the image subtraction algorithm. Doing so can also avoid the fitting being strongly misled by specific regions of the field (e.g., the brightest and densest regions).

Very recently, Hitchcock et al. (2021) provided an innovative way (i.e., **PyTorchDIA**) to bypass the construction of the analytical least-squares matrix. **PyTorchDIA** finds the kernel solution by optimizing a loss function using the automatic differentiation in PyTorch, which brings a

⁴ <https://github.com/thomasvrussell/sfft>

considerable gain in computing cost compared with that in Bramich (2008). However, PyTorchDIA is incompatible with kernel spatial variation and can only solve a constant kernel for image subtraction. In addition to the efforts on kernel determination, Hartung et al. (2012) successfully implemented fast image convolution with a spatially varying kernel on GPUs, which helps accelerate the subtraction after the convolution kernel has been computed.

The strategy of SFFT is different from the methods mentioned above: we present the least-squares question in Fourier space. By Parseval's theorem, a least-squares minimization in real space is equivalent to a least-squares minimization in the Fourier domain. Given that δ functions have higher flexibility and can maintain simple forms after Fourier transform, here we adopt the DBFs as the kernel basis functions following Bramich (2008) and Miller et al. (2008). Finally, SFFT allows for spatial variations across the field, whether from PSF, photometric scaling, or sky background. The following section will show how the calculations for image subtraction can be reduced to be FFTs and element-wise matrix operations.

2.2. Derivation of Subtraction

Given a reference image R and a science image S , PSF homogenization is carried out by convolving R or S with a spatially varying kernel $K_{x,y}$. Considering that the sky background between the two epochs may change, we introduce an additional term B to model the sky background difference between the two images. The image subtraction problem can be written in real space as the minimization of the difference image D , defined as

$$\begin{aligned} D(x, y) &= S(x, y) - \iint dudv R(x - u, y - v) K_{x,y}(u, v) \\ &\quad - B(x, y) \\ &= S(x, y) - (R \otimes K)(x, y) - B(x, y), \end{aligned} \quad (1)$$

where the input images R and S are images with dimensions (N_0, N_1) , and x and y are indices in the ranges of $[0, N_0 - 1]$ and $[0, N_1 - 1]$, respectively. Note that the kernel $K_{x,y}$ is spatially variant, so the integral in the equation above, strictly speaking, is not a convolution, such an integral is denoted by $R \otimes K$. The differential background map B is modeled as a polynomial form following Alard & Lupton (1998), that is,

$$B(x, y) = \sum_{pq} b_{pq} x^p y^q, \quad (2)$$

where the polynomial power indices p and q are in the range of $[0, D_B]$ and $[0, D_B - p]$, respectively. We follow Miller et al. (2008) and Hartung et al. (2012) to decompose the kernel $K_{x,y}$ into the *shape-free* δ -function basis $\mathcal{K}$. The kernel dimension is assumed to be (L_0, L_1) , with L_0 and L_1 being odd integers, such that $L_0 = 2w_0 + 1$ and $L_1 = 2w_1 + 1$. We assume

$$\mathcal{K}_{00}(u, v) = \mathcal{D}_{00}(u, v) = \vec{\delta}(u, v), \quad (3)$$

and

$$\begin{aligned} \mathcal{K}_{\alpha\beta}(u, v) &= \mathcal{D}_{\alpha\beta}(u, v) - \mathcal{D}_{00}(u, v) \\ &= \vec{\delta}(u - \alpha, v - \beta) - \vec{\delta}(u, v), \end{aligned} \quad (4)$$

where $\mathcal{D}_{\alpha\beta}(u, v) = \vec{\delta}(u - \alpha, v - \beta)$ is the standard Cartesian orthonormal basis, which consists of δ functions at each kernel

pixel, α and β are indices in the range of $[-w_0, w_0 + 1)$ and $[-w_1, w_1 + 1)$, respectively, u and v are kernel coordinate indices in the range of $[-w_0, w_0 + 1)$ and $[-w_1, w_1 + 1)$, respectively, and $\vec{\delta}$ is a binary function on integers $\vec{\delta}(\rho, \epsilon) = 1$ if $\rho = \epsilon = 0$, otherwise $\vec{\delta}(\rho, \epsilon) = 0$ with ρ and ϵ being any integers. With such a construction, the sum of the convolution kernel is uniquely determined by the coefficient of the basis vector $\mathcal{K}_{00}$, which simplifies the way we control the photometric scaling through convolution.

The spatial variation of the kernel can be fitted by the sum of the kernel function given above, modulated by a two-dimensional polynomial function to account for their varying contributions across the image field,

$$K_{x,y} = \sum_{\alpha\beta} \hat{A}_{xy\alpha\beta} \mathcal{K}_{\alpha\beta}, \quad (5)$$

$$\hat{A}_{xy\alpha\beta} = \sum_{ij} \hat{a}_{ij\alpha\beta} x^i y^j, \quad (6)$$

where the polynomial power indices i and j are in the range of $[0, D_K]$ and $[0, D_K - i]$, respectively. The information on the spatial variation of the kernel is encoded in the coefficients $\hat{a}_{ij\alpha\beta}$, which will be eventually solved by the subsequent linear system during the minimization of the residuals. More specifically, once $\hat{a}_{ij\alpha\beta}$ is known, we can calculate $\hat{A}_{xy\alpha\beta}$ for any given pixel coordinate (x, y) following Equation (6), and further derive the certain kernel associated with the pixel via the expansion on a δ -function basis by Equation (5).

Equations (2), (5), and (6) can be replaced by more complex functions. The code we developed includes an option for using B-splines to model the PSF and background level variations. For the B-splines cases, the space-varying kernel $K_{x,y}$ and differential background $B(x, y)$ are modeled as

$$K_{x,y} = \sum_{\alpha\beta} \hat{A}_{xy\alpha\beta} \mathcal{K}_{\alpha\beta}, \quad (7)$$

$$\hat{A}_{xy\alpha\beta} = \sum_{ij} \hat{a}_{ij\alpha\beta} B_{i;k,t}(x) B_{j;k,t}(y), \quad (8)$$

and

$$B(x, y) = \sum_{pq} b_{pq} B_{p;k',t'}(x) B_{q;k',t'}(y), \quad (9)$$

where $B_{i;k,t}$ and $B_{j;k,t}$ ($B_{p;k',t'}$ and $B_{q;k',t'}$) are B-spline basis functions of degree k (k') and knots t (t'). For simplicity, in this paper we focus only on the performance with polynomial models.

The pixel-to-pixel flux variations of the two images are accounted for by the coefficients $\hat{A}_{xy00} = \sum_{ij} \hat{a}_{ij00} x^i y^j$. If we consider the flux level of the image pair to be well calibrated, the constant flux scaling between images requires a constant kernel integral, that is, $\hat{A}_{xy00} = \hat{a}_{0000}$. Note that a constant flux scaling was first presented in Alard & Lupton (1998). Having a constant photometric ratio across the entire field is optional in our program. Like the method in Bramich et al. (2013), SFFT allows for space-varying flux scaling with a polynomial form to accommodate the effect of imperfect flat-field correction or cirrus cloud attenuation. Note that the current SFFT does not disentangle the polynomial degrees of spatial variations accounting for the convolution kernel and the photometric sensitivity.

With the abbreviation $T^{\rho\epsilon}(x, y) = x^\rho y^\epsilon$ and $R^{\rho\epsilon} = T^{\rho\epsilon}R$, where ρ and ϵ are any integers, Equation (1) is rewritten as

$$D(x, y) = S(x, y) - \sum_{ij\alpha\beta} \hat{a}_{ij\alpha\beta} [T^{ij}(R \otimes \mathcal{K}_{\alpha\beta})](x, y) - \sum_{pq} b_{pq} T^{pq}(x, y). \quad (10)$$

The convolution kernel is typically very small in size, at such scale its spatial variation is expected to be negligible locally. An approximation (see Appendix A) can be made by moving the polynomial functions describing the spatial variation outside the integral in Equation (1). This leads to

$$D(x, y) = S(x, y) - \sum_{ij\alpha\beta} \hat{a}_{ij\alpha\beta} (R^{ij} \otimes \mathcal{K}_{\alpha\beta})(x, y) - \sum_{pq} b_{pq} T^{pq}(x, y), \quad (11)$$

and

$$D(x, y) = S(x, y) - \sum_{ij\alpha\beta} \hat{a}_{ij\alpha\beta} R^{ij} \circ \mathcal{K}_{\alpha\beta} - \sum_{pq} b_{pq} T^{pq}. \quad (12)$$

Note the notation $\circ$ in Equation (12) indicates circular convolution and $\mathcal{K}_{\alpha\beta}$ can be estimated to generate the convoluted source image that best matches the PSF of the reference image across the entire field.

The δ -function basis we have adopted for $\mathcal{K}$ as shown in Equations (3) and (4) allows for a simple Fourier space representation of the image difference procedure. In Fourier space, Equation (12) becomes

$$\hat{D} = \hat{S} - \sum_{ij\alpha\beta} a_{ij\alpha\beta} \widehat{R^{ij} \mathcal{K}_{\alpha\beta}} - \sum_{pq} b_{pq} \widehat{T^{pq}}, \quad (13)$$

where the symbols with a hat denote the Fourier transform of the images, $a_{ij\alpha\beta} = N_0 N_1 \hat{a}_{ij\alpha\beta}$ with N_0 and N_1 being the dimensional of the images in x and y , respectively. The discrete Fourier transform (DFT) of the basis function has the simple form

$$\begin{aligned} \widehat{\mathcal{K}_{\alpha\beta}}(l, m) &= \frac{1}{N_0 N_1} (W_{N_0}^{l\alpha} W_{N_1}^{m\beta} - 1) \text{ if } \alpha \text{ or } \beta \neq 0 \\ &= 1/N_0 N_1 \text{ if } \alpha = \beta = 0, \end{aligned} \quad (14)$$

where $W_{N_0} = e^{-i2\pi/N_0}$ and $W_{N_1} = e^{-i2\pi/N_1}$, with i being the unitary imaginary number.

Now we define $G = \hat{D} \hat{D}^*$, where $*$ stands for a complex conjugate, and we find

$$\begin{aligned} G &= 2\Re(\hat{S}) - 2 \sum_{ij\alpha\beta} a_{ij\alpha\beta} \Re[\hat{S}^* \widehat{R^{ij} \mathcal{K}_{\alpha\beta}}] \\ &+ 2 \sum_{ij\alpha\beta} \sum_{pq} a_{ij\alpha\beta} b_{pq} \Re[\widehat{R^{ij} \mathcal{K}_{\alpha\beta}} (\widehat{T^{pq}})^*] \\ &+ \sum_{ij\alpha\beta} \sum_{i'j'\alpha'\beta'} \alpha' \beta' a_{ij\alpha\beta} \widehat{R^{ij} \mathcal{K}_{\alpha\beta}} a_{i'j'\alpha'\beta'} (\widehat{R^{i'j'}})^* (\widehat{\mathcal{K}_{\alpha'\beta'}})^* \\ &- 2 \sum_{pq} b_{pq} \Re[\hat{S}^* \widehat{T^{pq}}] \\ &+ \sum_{pq} \sum_{p'q'} p' q' b_{pq} \widehat{T^{pq}} b_{p'q'} (\widehat{T^{p'q'}})^*. \end{aligned} \quad (15)$$

Our goal of optimal subtraction is to minimize the power of the difference image in Fourier space, as given in Equation (15). Let $F = \sum_{lm} G(l, m)$, the minimization in Fourier space for image subtraction has a simple least-squares

solution given by $\nabla F = 0$ where the gradients are calculated with respect to all the parameters for image matching,

$$\begin{aligned} \partial F / \partial a_{ij\alpha\beta} &= \sum_{lm} \{-2\Re[\hat{S}^* \widehat{R^{ij} \mathcal{K}_{\alpha\beta}}] \\ &+ 2 \sum_{ij\alpha\beta} a_{ij\alpha\beta} \Re[(\widehat{R^{ij}})^* (\widehat{\mathcal{K}_{\alpha\beta}})^* \widehat{R^{ij} \mathcal{K}_{\alpha\beta}}] \\ &+ 2 \sum_{pq} b_{pq} \Re[\widehat{R^{ij} \mathcal{K}_{\alpha\beta}} (\widehat{T^{pq}})^*]\}(l, m) \\ &= 0, \end{aligned} \quad (16)$$

$$\begin{aligned} \partial F / \partial b_{pq} &= \sum_{lm} \{-2\Re[\hat{S}^* \widehat{T^{pq}}] \\ &+ 2 \sum_{ij\alpha\beta} a_{ij\alpha\beta} \Re[(\widehat{R^{ij}})^* (\widehat{\mathcal{K}_{\alpha\beta}})^* \widehat{T^{pq}}] \\ &+ 2 \sum_{pq} b_{pq} \Re[(\widehat{T^{pq}})^* \widehat{T^{pq}}]\}(l, m) \\ &= 0. \end{aligned} \quad (17)$$

Equations (16) and (17) form a linear system with the array elements shown in Equation (18).

$$\begin{bmatrix} A_{ij\alpha\beta ij\alpha\beta} & B_{ij\alpha\beta pq} \\ \vdots & \vdots \\ \tilde{B}_{pq ij\alpha\beta} & C_{pq pq} \end{bmatrix} \begin{Bmatrix} a_{ij\alpha\beta} \\ \vdots \\ b_{pq} \end{Bmatrix} = \begin{Bmatrix} D_{ij\alpha\beta} \\ \vdots \\ E_{pq} \end{Bmatrix} \quad (18)$$

The elements of Equation (18) are given explicitly as follows:

$$A_{ij\alpha\beta ij\alpha\beta} = \sum_{lm} \{\Re[\widehat{R^{ij} \mathcal{K}_{\alpha\beta}} (\widehat{R^{ij}})^* (\widehat{\mathcal{K}_{\alpha\beta}})^*]\}(l, m), \quad (19)$$

$$B_{ij\alpha\beta pq} = \sum_{lm} \{\Re[\widehat{R^{ij} \mathcal{K}_{\alpha\beta}} (\widehat{T^{pq}})^*]\}(l, m), \quad (20)$$

$$\tilde{B}_{pq ij\alpha\beta} = \sum_{lm} \{\Re[\widehat{T^{pq}} (\widehat{R^{ij}})^* (\widehat{\mathcal{K}_{\alpha\beta}})^*]\}(l, m), \quad (21)$$

$$C_{pq pq} = \sum_{lm} \{\Re[\widehat{T^{pq}} (\widehat{T^{pq}})^*]\}(l, m), \quad (22)$$

$$D_{ij\alpha\beta} = \sum_{lm} \{\Re[\hat{S}^* \widehat{R^{ij} \mathcal{K}_{\alpha\beta}}]\}(l, m), \quad (23)$$

$$E_{pq} = \sum_{lm} \{\Re[\hat{S}^* \widehat{T^{pq}}]\}(l, m). \quad (24)$$

These equations can be further simplified as shown in Appendix B. Finally, the difference image D is calculated by Equation (13) by applying the linear system solution $\{..., a_{ij\alpha\beta}, ..., b_{pq}, ...\}$ of Equation (18). Given that the difference image will possess correlated noise on the background, we present a decorrelation process to whiten the background noise in Appendix C.

3. Implementation of SFFT

We have presented the mathematical derivation of the SFFT image difference algorithm in Section 2.2. In a nutshell, to

calculate the difference image D from a pair of the reference image (R) and science image (S), one can first search for the optimal parameters $\{..., a_{ijab}, ..., b_{pq}, ...\}$ by minimizing the power of the resulting difference image of R and S in Fourier space via the linear system described in Equation (18). Subsequently, one needs to apply the solution to match image R to image S following Equation (13) and get the ultimate difference image D .

However, saturated sources with a significant nonlinear response, casual cosmic rays and moving objects, bad CCD pixels, optical ghosts, and even the variable objects and transients themselves can severely affect the construction of reliable convolution kernels. These objects can be seen as distractions for image subtraction (hereafter referred to as *distraction sources*), which are hardly modeled by the image subtraction algorithm but are ubiquitous in real observations. We further need to offer an effective channel to prevent the *distraction sources* in R and S from contributing to the parameter-solving process. One likely choice is to deweight those trivial pixels, but it appears to be tricky to formulate the weight assignment in Fourier space. Instead, we found that a more straightforward approach by adequate image masking is easier to implement and can perform well enough in our study.

In the current implementation of SFFT, image subtraction is developed as a two-step process. First, make a masked version of reference image R and science image S , denoted as $\tilde{R}$ and $\tilde{S}$, respectively. Second, solve the linear system in Equation (18) using the masked image pair $\tilde{R}$ and $\tilde{S}$, then apply the solution to calculate the difference image D of the original image pair R and S following Equation (13). The SFFT algorithm does not rely on isolated stellar objects to construct PSF-matching kernels, thereby performing equally well for the observations taken in crowded and sparse fields. Our software provides two flavors for image subtraction, *crowded-flavor* and *sparse-flavor*, to accommodate the situations for both crowded and sparse fields. But the flavor for crowded fields can also work for sparse fields, except the results may be affected by the large number of noisy background pixels that do not contribute to the construction of the convolution kernel. The two flavors follow the same routine for image subtraction and differ only in ways of masking the data.

3.1. Preprocessing for Crowded-flavor SFFT

Applying SFFT is straightforward for signal-dominated crowded fields. Mostly, saturation is the predominant factor in all *distraction sources* for crowded fields. The crowded-flavor SFFT will automatically mask the pixels affected by the saturated sources in the field and replace them with the neighboring background. To further reduce image matching errors, it is allowed to use a more elaborate customized mask instead, e.g., to include the pixels affected by known variables and transients in the field.

3.2. Preprocessing for Sparse-flavor SFFT

On the other hand, sparse-flavor SFFT is relatively more sophisticated in image masking. We select a set of sources of astronomical significance in the field and mask all other irrelevant regions so that the selected objects can dominate the parameter-solving process. It is reminiscent of the similar strategy in HOTPANTS, which fits on the rectangle stamps (i.e., *sub-stamps* in HOTPANTS terminology) encompassing selected astronomical objects. The analogous intention of both methods is to restrict the calculation on a properly preselected set of

sources to eliminate the effect of *distraction sources* on the solution of convolution kernels. In this scheme, the regions masked by SFFT also include abundant background pixels. This is a reasonable consideration under our framework. Pixel uncertainty is not considered as weights in SFFT subtraction. Accordingly, the overwhelming noise-dominated background pixels are more likely to degrade the construction of accurate PSF-matching kernels rather than to contribute to it (see the same consideration in Kochanski et al. 1996).

In practice, the input image pair of sparse-flavor SFFT is required to be sky subtracted. This requirement is to simplify the image-masking process so that all the pixels enclosed in masked regions can be replaced by a constant of zero. As a result, the differential background term in the SFFT algorithm becomes trivial as the background offset between the input image pair has been minimized by sky subtraction. That is to say, we pass on the function of matching differential background, which was embedded in the numerical calculations of SFFT, to a customized sky subtraction as a preliminary process. Fortunately, modeling sky background is usually feasible in sparse fields (e.g., using an interpolation-based method). Note this requirement is not a prerequisite for crowded-flavor SFFT. This is because properly modeling sky background can be tricky for crowded fields: the modeled sky is more susceptible to being biased by the signal harboring in the pixels misidentified as background (e.g., the outskirts of nearby galaxies, see Akhlaghi & Ichikawa 2015).

We developed a morphological classifier to carry out the source selection in sparse-flavor SFFT. The classifier was initially proposed for PSFEX (Bertin 2011), which enables PSFEX to select a subset of point sources for constructing the PSF model. Given a photometry catalog generated by SEXTRACTOR (Bertin & Arnouts 1996), one can draw a figure of instrumental magnitudes (SEXTRACTOR catalog value MAG_AUTO) against flux radius (SEXTRACTOR catalog value FLUX_RADIUS) for all detected photometry objects. Generally, bright point sources tend to stay around a nearly vertical straight line on this figure. PSFEX leverages the statistical feature to select appropriate samples to establish the PSF model. Although the goal of source selection for sparse-flavor SFFT is not entirely aligned with that for PSFEX, it inspired us to make the selection based on SEXTRACTOR parameters and their statistical characters.

To demonstrate the source selection criterion in sparse-flavor SFFT, we show an example of an individual DECam (Honscheid & DePoy 2008) image in Figure 1. By contrast, the SEXTRACTOR photometry catalog of this image is cross matched with the Tractor catalog from the Legacy Survey (Blum et al. 2016), which offers its own fitted morphological types. As shown in Figure 1, the point sources and extended sources form two conspicuous branches, which intersect at the faint end but become well separated toward the bright side. According to the Tractor types, the branch with a nearly vertical orientation primarily comprises point-like sources with the Tractor type PSF, while the extended sources with Tractor types DEV, EXP, and REX are likely to be found in the other branch that sprawls out horizontally. Moreover, the discrete gray dots represent the objects that cannot be found in the Tractor catalog, which are casual detections such as cosmic rays. SFFT uses Hough transformation (Hough 1959) to identify the straight line surrounded by the branch of point sources. The figure is first pixelized by counting the number of objects in each small grid. Hough transformation is applied so

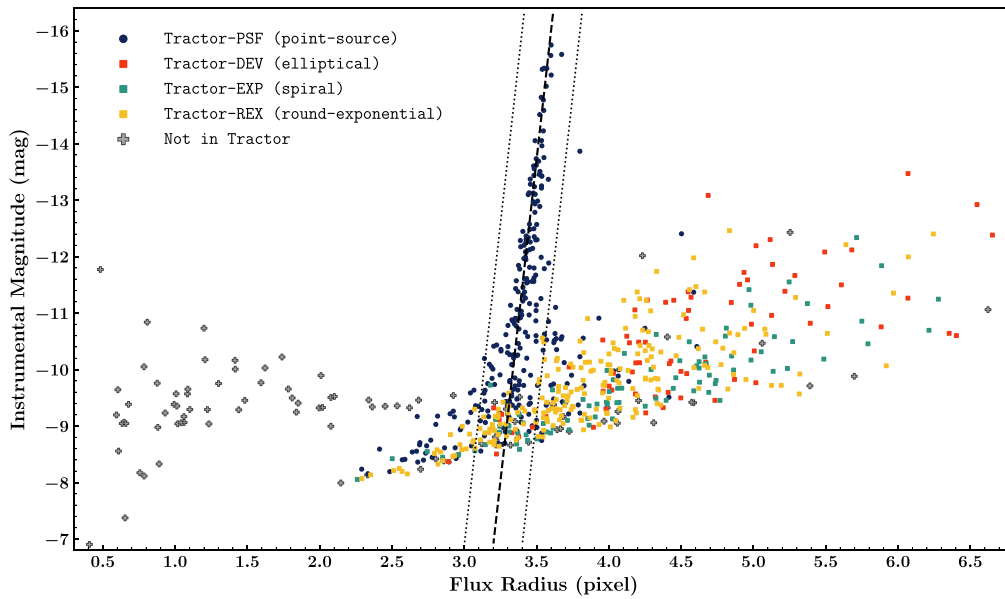


Figure 1. Morphological classifier in sparse-flavor SFFT demonstrated by an arbitrary DECam image of Legacy Survey (Blum et al. 2016). The classifier is based on the photometric results of the image using *SEXTRACTOR*. The data points show the relationship between instrumental magnitude (*SEXTRACTOR* catalog value *MAG_AUTO*) and flux radius (*SEXTRACTOR* catalog value *FLUX_RADIUS*) for the detected objects with *SEXTRACTOR* catalog value *FLAG* being zero. We also group the data by the morphological types in the Tractor catalog of Legacy Survey (Blum et al. 2016) shown with different markers in the figure. The dashed black line is detected by Hough transformation as the strongest straight line feature with vertical orientation. Two dotted lines are parallel to the dashed line with fixed separation being 0.2 (SFFT parameter *BeltHW*), the region between that is referred to as the point-source belt.

that we can search for the strongest line feature with vertical orientation. The thin belt-like region around the detected straight line with a fixed width is referred to as the *point-source belt*. In SFFT, the parameter *BeltHW* describes the belt half-width (default value is 0.2). With this terminology, it is easier to describe the specific selection criteria.

For the source selection, sparse-flavor SFFT first generates a *SEXTRACTOR* photometry catalog for reference image and the science image, respectively. A source is selected if it lies in the point-source belt or out of the point-source belt on the right side for both images. The extended sources in the field are not excluded, as SFFT can use both stars and galaxies to derive the solution for the matching kernel. However, typical cosmic rays and the faintest astronomical sources will be discarded. Given that each pixel of the selected samples, whether bright or faint, is equally weighted in SFFT, rejecting the noisiest subset should be appropriate under our framework. Furthermore, *SEXTRACTOR* has been configured to only output the objects with the *SEXTRACTOR* catalog value *FLAG* of zero. Namely, this guarantees that the saturated sources and blending objects (relatively rare in sparse fields) are not in the selection.

Significant variable sources in the field should be further rejected. SFFT cross matches the photometry catalogs of the input image pair, then calculates the difference in instrumental magnitude for each matched object. The outliers of the distribution of magnitude difference indicating a violent change in brightness will be discarded from the initial set of selection. In SFFT, the outliers are identified by the threshold parameter *MAGD_THRESH* (the default value is 0.12 mag). The surviving selection set is applied to the image masking in sparse-flavor SFFT. The rejection of an outlier is usually sufficient enough for a successful image subtraction, but one is also allowed to further refine the automatic selection by removing the known but not recognized variables and transients. Unlike *HOTPANTS*, SFFT does not enclose each source of the selection within a fixed-sized rectangle stamp.

Instead, we employ the *SEXTRACTOR* segmentation map to define the pixel domain of each selected source. Finally, the pixels that are not in any pixel domain will be masked by a constant of zero.

3.3. Implementation of SFFT Subtraction

The next step is to perform subtraction with the unmasked and the masked image pairs, in the same way for the two flavors. We implemented the subtraction algorithm in CUDA following the mathematical reasoning in Section 2.2.

There are a few free parameters in this process. In Section 2.2, the spatial variation of the convolution kernel was modeled as a polynomial form of degree D_K . Similarly, the differential background is assumed to be a polynomial with degree D_B . In SFFT, the two free parameters correspond to *KerPolyOrder* and *BGPolyOrder*, respectively. The default values of the two parameters are 2. Another parameter that should be specified is the pixel size of the convolution kernels. The default way is to determine kernel size as seeing related, i.e., the kernel half-width is calculated as $\text{KerHWRatio} \times \text{FWHM}_L$, where the ratio *KerHWRatio* is a free parameter of SFFT and FWHM_L is the worst seeing of the input image pair. The last free parameter *ConstPhotRatio* is boolean, which controls whether SFFT subtraction is subject to a constant photometric ratio or not. The default value of this parameter is *True*, which assumes the input image pair has been well calibrated. Setting the parameter to *False* means the photometric ratio will also become a polynomial form of degree *KerPolyOrder*.

4. Examples of Image Subtraction with SFFT

We show examples of SFFT applied to observations from five different telescopes. These data correspond to a diverse range of characteristics. The technical specifications of the instruments are given in Table 1. To test the two flavors of SFFT, we use ZTF (Bellm et al. 2019), AST3-II (Yuan et al. 2014),

Table 1
Technical Specifications of the Instruments

Instrument	Telescope (m)	Pixel Scale (arcsec pixel ⁻¹)	Field of View (deg ²)	Field Property	Comments
ZTF	1.2	1.0	47.0	Crowded	M31
AST3-II	0.5	1.0	4.3	Crowded	LMC
TESS	4 × 0.1	21.0	4 × 576.0	Crowded	Space based
DECam	4.0	0.262	3.0	Sparse	...
TMTS	4 × 0.4	1.86	4 × 4.5	Sparse	CMOS detectors

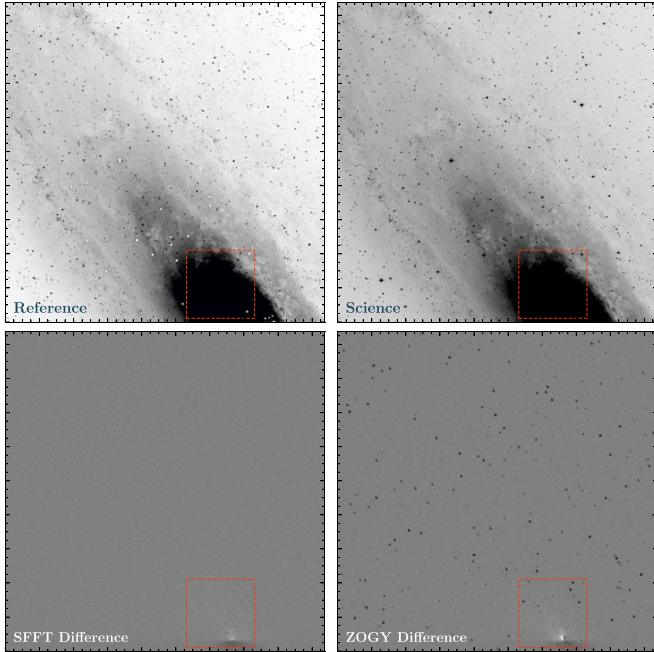


Figure 2. Image subtraction performance of the ZTF test with observations covering M31. The upper panel, from left to right, shows the input reference image *ZTF-REF* and science image *ZTF-SCI*. The lower panel presents the resulting difference image from SFFT (left) and *ZOGY* (right), where the *ZOGY* difference is directly downloaded from ZTF Data Release 3. The red dashed square marks the core region of M31.

and TESS (Ricker et al. 2015) data as examples of crowded fields. Also, the observations from DECam and TMTS with abundant isolated stars are selected as representatives of sparse fields.

The ZTF (Bellm et al. 2019) and AST3-II (Wang et al. 2017; Yuan et al. 2014) data are images of the nearby galaxies M31 and the Large Magellanic Clouds (LMC), respectively. The two wide-field survey telescopes have the same pixel sampling but point to a different hemisphere. We additionally test SFFT with TESS (Ricker et al. 2015) images with a vastly different pixel scale of 21.0 arcsec pixel⁻¹ where PSF is severely under-sampled. All of these selected observations could be seen as signal-dominated cases so that subtraction tests on them are conducted by the crowded-flavor SFFT.

The DECam and TMTS data were acquired from extragalactic transient surveys (Mould et al. 2017; Zhang et al. 2020). The DECam data are from deep surveys with a large number of isolated point sources and extended galaxies, while the TMTS data are from a shallow nearby survey that recorded plenty of point sources. TMTS has a larger pixel scale than DECam, but its PSF is not under-sampled, unlike TESS. For these observations, the observed stellar objects are generally

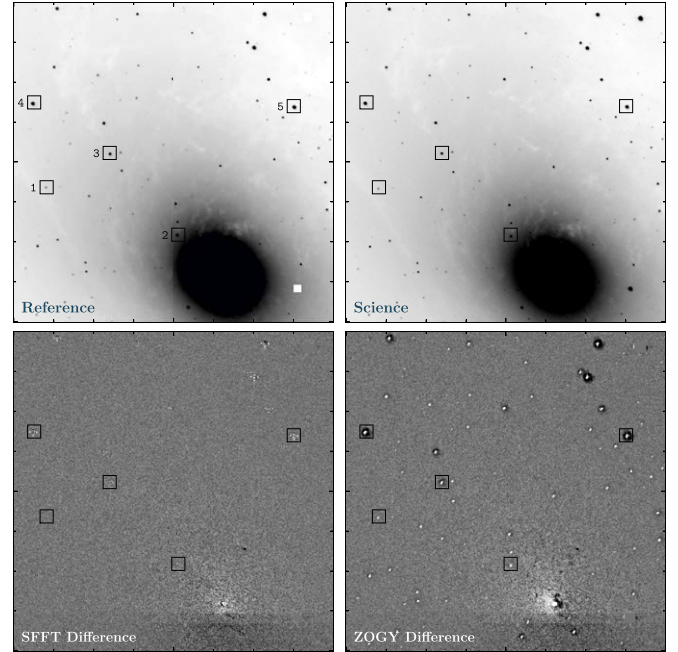


Figure 3. Image subtraction performance of the ZTF test around the galaxy center of M31. The panel descriptions are the same as those in Figure 2, however, in a close-up view of the region enclosed by the red square in Figure 2. In each panel, the solid black squares mark five field stars to be checked in detail as shown in Figure 4.

isolatedly distributed in the field, making it possible to apply the source selection described in Section 3. We use sparse-flavor SFFT for the subtraction tests on DECam and TMTS.

Throughout this section, SFFT always uses the default configuration of the free parameters described in Section 3, which are summarized in Table D1, even though the two sets of images differ drastically. Detailed information on the test data used in this section is presented in Table E1. In this context, we will highlight the alias names of the test images in italics. All input image pairs for image subtraction have been registered by *SWarp* (Bertin 2010) with the LANCZOS3 resampling function. For the input images of sparse-flavor SFFT, we have modeled and subtracted their sky using *SEXTRACTOR* with a mesh size of 64 × 64 pixels.

4.1. Crowded Fields

By design, SFFT does not require isolated sources to solve the image matching. In crowded fields where few or no isolated stars are present, SFFT can perform exceptionally well. The ZTF test images are downloaded from ZTF Data Release 3,⁵ covering the nearby galaxy M31. Figures 2 and 3 show the subtraction performance of both the SFFT and *ZOGY* method,

⁵ <https://www.ztf.caltech.edu/page/dr3>

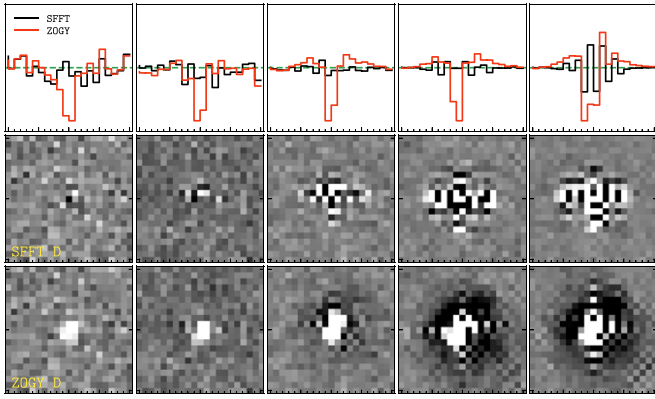


Figure 4. Analysis of flux residuals of the ZTF test. The second row and third row show thumbnail images of the selected five samples from the SFFT difference and ZOGY difference, respectively. From left to right, thumbnail images are sorted by the sample labels shown in Figure 3. For each thumbnail image, the flux integral along the y -axis can produce a flux curve as a function of the index of the x -axis. These functions are shown in the first row, denoted by red solid curves for ZOGY and black solid curves for SFFT. In each panel of the first row, the two flux curves have been rescaled with the same factor for the clarity of the display clarity. The green dashed line denoted the flux level of constant zero.

with a wider view and a close-up view of the galaxy core, respectively.

The difference image generated by SFFT is much cleaner than that from the ZOGY approach, as indicated by the lower panels of the two figures. One can also notice that the difference image by the ZOGY method appears to have a bias seemingly due to the photometric mismatch of image subtraction. This guess is confirmed in Figure 4, which illustrates an analysis of the flux residuals of five field stars near the galaxy core. ZOGY subtraction uniformly remains negative net flux for these samples, suggesting an improperly estimated photometric ratio between the reference and science images. By contrast, SFFT does not have the same problem. Unlike ZOGY, the photometric ratio (a constant in our default configuration, i.e., $\hat{a}_{0000}$) in SFFT is straightforwardly solved from the linear system, rather than an extra value calculated by some other photometric process.

Figures 5 and 6 show the SFFT subtraction performance on LMC observations of the Antarctica telescope AST3-II and test data from space-based TESS 30 minute cadence full frame images (FFIs), respectively. A known variable star of RRAB type is easily identified at the center of the image for both cases. It illustrates how the flux variability can be effectively captured by SFFT subtraction from the densely packed fields.

4.2. Sparse Fields

It is also essential to investigate the performance of SFFT subtraction on the ordinary sparse sky fields that are more common in extragalactic transient surveys. For comparison, we additionally employed the widely used image subtraction software HOTPANTS as a baseline method. As one of the specific implementations of Alard & Lupton (1998), HOTPANTS adopts Gaussian basis functions to construct the convolution kernel. In real space, the use of DBFs has been shown to yield better performance than Gaussian basis functions in terms of photometric accuracy (Albrow et al. 2009). Here we present the comparisons to HOTPANTS to confirm that SFFT, though working in Fourier space, will still

inherit the advantage of DBFs compared with Gaussian basis functions. The parameters of HOTPANTS used for this section are listed in Table D2. Recall that sparse-flavor SFFT restricts the calculations of convolution kernels on a preselected set of sources, this source list used by SFFT will be shared by HOTPANTS for the sake of fairness. Besides, both methods always apply a consistent size of convolution kernels and use the same degree of the polynomial of the spatial variation for image subtraction. We selected 25 DECam observations with the same pointing but with diverse seeing conditions from our DECamERON survey (Mould et al. 2017) (see Table E1). With the set of DECam images, we conduct two types of tests to check the cleanness of the subtraction and photometric accuracy, respectively.

For the subtraction cleanness test, we use the image *DECam-SREF* as the shared reference, having the best seeing among all of them. The remaining 23 images aliased by initial DECam-OBS are regarded as science images in this test. Image subtractions are performed for the sequence of science images with a shared reference using sparse-flavor SFFT and HOTPANTS, respectively. An example of the results of the subtraction is shown in Figure 7 from the observation *DECam-OBS02f*. For both methods, the calculations involved in parameter-solving have been restricted to the same preselected sources, and the minimization aims at optimal subtraction on these sources as clean as possible. It is useful to give a close-up view of the performance of the subtraction specifically for the selection. For each image in Figure 7, we cut a small thumbnail image around each selected source and combine the cutouts into a grid to be a new synthetic image, as shown in Figure 8. Note that the source selection, in this case, is more than 1100, so we only exhibit a subset of size 64 from the bright end of the complete set for clarity of display. One can notice that the most conspicuous subtraction-induced artifacts have a dipole-like pattern that is especially prominent for the bright sources. The pattern can be found in both difference images, but is much less profound in SFFT subtractions than in HOTPANTS subtractions. It is an expected improvement as the SFFT kernel solution with DBFs allows more degrees of freedom than HOTPANTS using Gaussian basis functions.

To further verify such an improvement is real in a statistical sense, here we define a naive quantifiable metric to describe the cleanness of the subtraction and show comparisons of the performance with the test data. Note that the subtraction-induced artifacts on difference images are usually detectable by SEXTRACTOR like other real transient and variable objects. Generally, a huge number of artifacts in transient surveys will survive until an AI-based stamp classifier recognizes them. This fact makes it possible to use a simple dichotomous metric to describe the cleanness of the subtraction. A given source is either clean-subtracted or not according to the existence of SEXTRACTOR detection of its subtraction residuals. Now we can investigate how the probability of clean subtraction for a given set of sources is related to the employed image subtraction method. For transient surveys, it is essential to probe the performance of the subtraction over the galaxies in the field, which are potential hosts of transient events. When a transient emerges at a position very close to its host galaxy core, the subtraction-induced artifacts from the galaxy can severely hinder the discovery and photometric measurement of the transient.

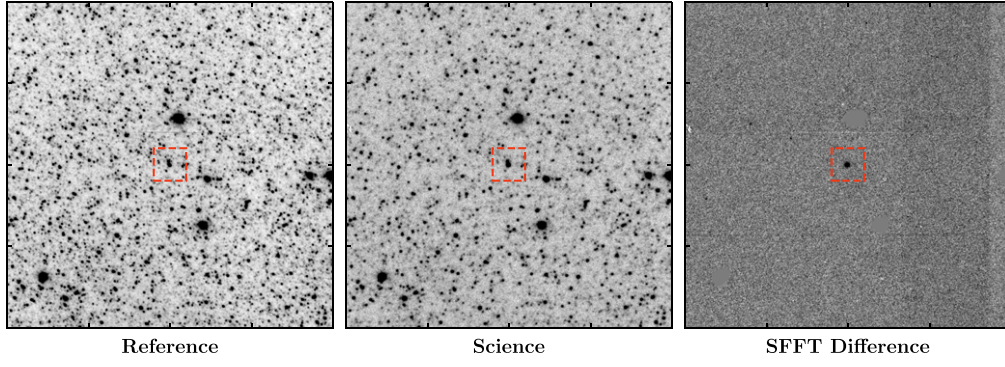


Figure 5. Image subtraction performance of the AST3-II test with observations of the LMC center. The figure, from left to right, shows the input reference image *AST-REF*, science image *AST-SCI*, and the resulting difference image by SFFT. The red dashed square at the center of the image indicates a known RRAB variable TY Dor (R.A. = 05:24:06.35 decl. = -69:25:11.0) from the VSX catalog (Watson et al. 2006). The vertical strip illustrates the detection readout sections with slightly different gain values.

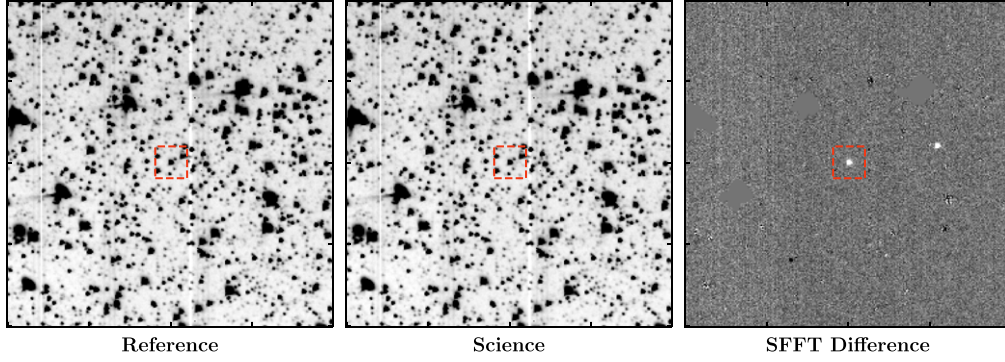


Figure 6. Performance of the image subtraction of the TESS test. The figure, from left to right, shows the input reference image *TESS-REF*, science image *TESS-SCI*, and the resulting difference image by SFFT. The red dashed square at the center of the image indicates a known RRAB variable CV Scl (R.A. = 23:09:30.52 decl. = -35:47:16.9) from the VSX catalog (Watson et al. 2006).

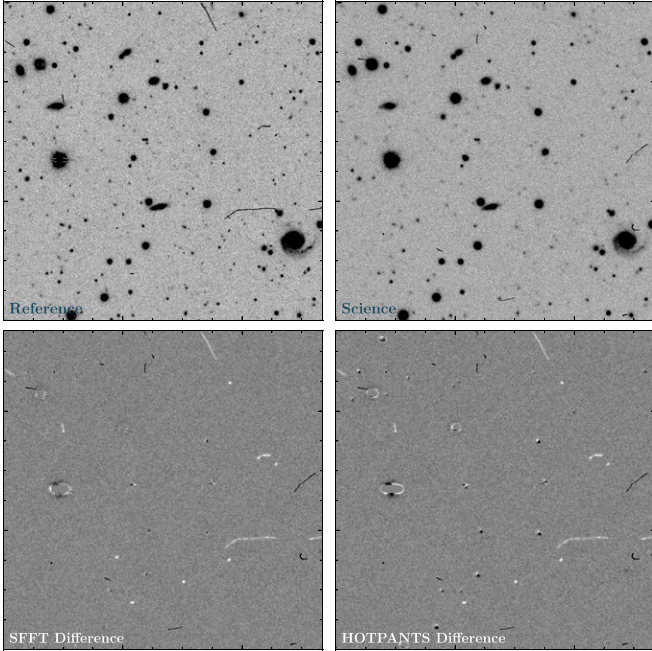


Figure 7. Image subtraction performance of the DECam test from the DECamERON survey. The upper panel shows the input reference image *DECam-SREF* and science image *DECam-OBS02f*. The lower panel shows the resulting difference image by SFFT (left) and HOTPANTS (right). The full field of view of DECam is too large for display, so here we only present them in a narrow view of a 3' width.

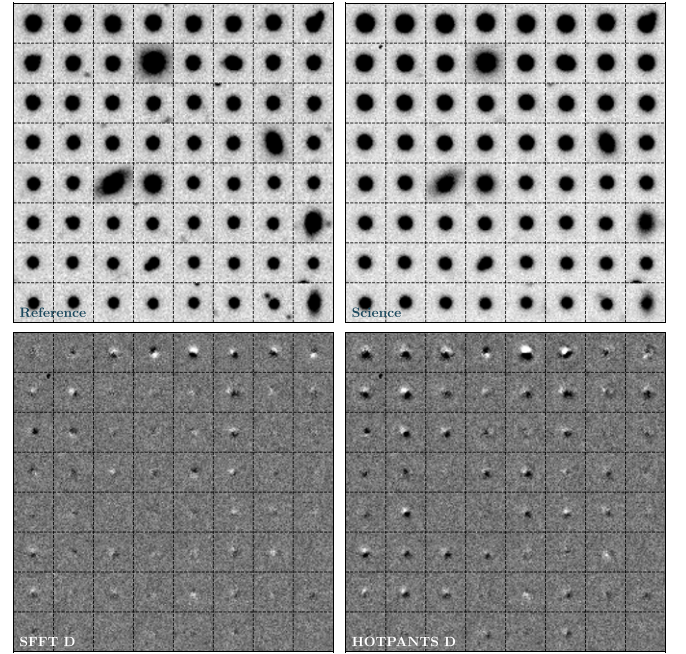


Figure 8. Performance of the image subtraction of the DECam test on the selected sources. The panels are arranged as in Figure 7 but show the subtraction results on the set of preselected individual sources. Each panel shows a synthetic image that combines thumbnail cutouts of 64 sources from the original images. The presented samples are drawn from the brightest end of the complete selection, and the thumbnails in the four panels are placed in order of decreasing brightness.

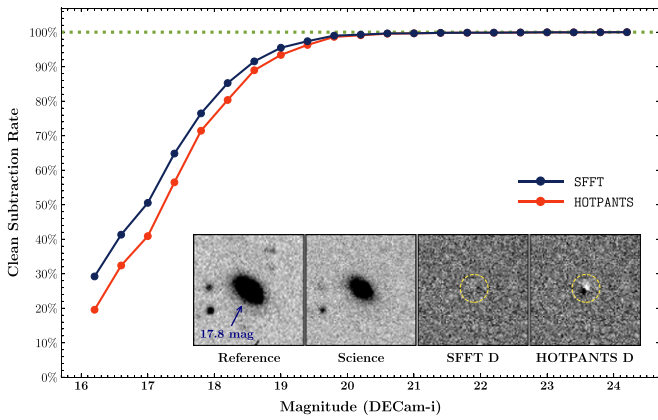


Figure 9. Clean subtraction rate of galaxies measured from the DECam test. The solid curves show a clean subtraction rate as a function of galaxy magnitude for SFFT (blue) and HOTPANTS (red), with a magnitude bin width of 0.4 mag. The panel insets show examples of the performance of the subtraction for a bright galaxy. The first two panels of the insert show the zoomed-in region around the galaxy from the reference image *DECam-SREF* and the science image *DECam-OBS18a*, respectively. The last two panels give the resulting difference image by SFFT and HOTPANTS. The yellow dashed circle indicates the search region for subtraction-induced artifacts with a radius of 11 pixels. The example represents typical cases when SFFT is able to provide a clean subtraction but HOTPANTS failed.

Our deep DECam observations in the test (3 deg², down to 23.5 mag in the *i* band) have covered a vast number of galaxies with various morphological types. It is interesting to calculate the statistics of the probability of clean subtraction for those sources identified as extended galaxies (i.e., Tractor morphological types DEV, EXP, and REX) in the Legacy Survey (Blum et al. 2016). The check radius for searching subtraction-induced artifacts using SEXTRACTOR is specified as 11 pixels (~ 2 times the median FWHM for DECam) from the galaxy core. Figure 9 shows the rate of clean subtraction as a function of the magnitude of the examined galaxy. As expected, the brighter galaxies have a higher probability of leaving detectable subtraction-induced artifacts. However, the SFFT method is more likely to achieve clean subtraction than HOTPANTS.

For the photometric accuracy test, we use the image *DECam-PTAR* as the PSF target, which has a median-level seeing among all selected DECam observations. The remaining 23 images aliased by initial DECam-OBS are convolved to achieve a homogeneous PSF using sparse-flavor SFFT and HOTPANTS, respectively. We derive the light curves of the field stars by conducting photometry on the sequence of convolved images with a fixed aperture. The photometric accuracy measured by the light curves can work as a metric to assess the PSF-matching quality. We investigate the aperture photometry of the point sources classified as Tractor morphological type PSF. Figure 10 shows the rms error of the derived light curves as a function of the star brightness. As the photometric uncertainty from SFFT is systematically lower than that from HOTPANTS, especially at the bright end, the improvement in PSF matching is again confirmed.

Figure 11 shows an example with TMTS data. The subtraction artifacts generated by SFFT and HOTPANTS are not identical. HOTPANTS produces relatively circular and symmetric patterns, while SFFT shows more random residuals. This obviously originates from the different base kernel construction in SFFT and HOTPANTS. SFFT uses a δ -basis kernel, which is more flexible, while HOTPANTS utilizes

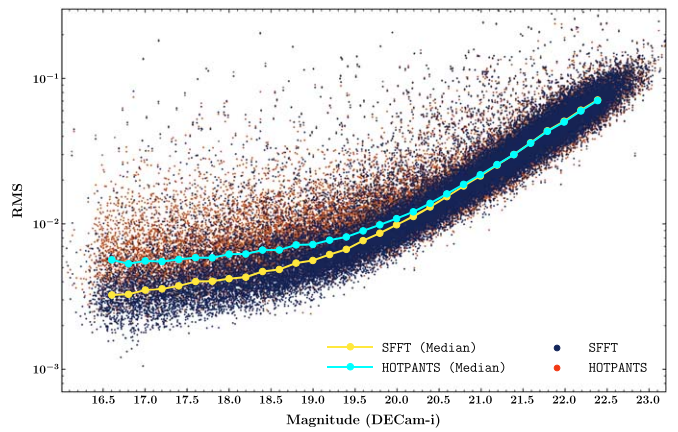


Figure 10. Photometric accuracy measured from light curves of field stars for DECam test. The light curve of each field star (classified as PSF in the Legacy survey) is derived by force aperture photometry on the queue of the convolved images (aliased with the initial DECam-OBS). The force photometry is carried out by SEXTRACTOR with Gaussian optimal aperture $1.3462 \times \text{FWHM}_{\text{PTAR}}$, where $\text{FWHM}_{\text{PTAR}}$ is the FWHM of the specific CCD tile of *DECam-PTAR*. The photometric uncertainty against star magnitude is shown as blue dots (for SFFT) and red dots (for HOTPANTS). The solid curves denote the median level of the rms in each magnitude bin with a width of 0.2 mag for SFFT (yellow) and HOTPANTS (cyan), respectively.

Gaussian functions, which possess a higher intrinsic symmetry about the kernel center.

5. Computational Performance of SFFT

The most remarkable feature of SFFT is that the image subtraction in Fourier space can be efficiently parallelized in GPU devices. With the simplification described in Appendix B, we have reduced the computationally intensive kernel determination as DFTs and element-wise matrix operations, which are exceptionally suitable for GPU graphic multiprocessors. This section will show the excellent computing performance of the SFFT algorithm in comparison with several existing image subtraction methods, including HOTPANTS, ZOGY,⁶ and PyTorchDIA.⁷ Note that the comparisons only focus on the limited specific implementations of image subtraction algorithms. One should not overinterpret the computing costs given in this section to represent the best possible computational performances of the general algorithms. Unless specified otherwise, the parameters of the software used in this section are listed in Tables D1–D4.

It is useful to demonstrate the computing performance of the image subtraction methods by testing them with a variety of data on different computational platforms. We select a set of testing input data with diverse image sizes, seeing conditions, and sky areas. The tests are performed on two computing servers with different hardware configurations. One server (denoted by I) is equipped with Intel(R) Xeon(R) CPU E5-2630 (2.20 GHz) and one NVIDIA Tesla V100 GPU, while the other server (denoted by II) has more advanced configurations with AMD EPYC CPU 7542 and one NVIDIA Tesla A100 GPU. Note that the time spent on preprocessing steps can also be considerable. We additionally record the computing cost of the pre-subtraction procedures in our tests. In this section, we conduct two types of tests using TMTS data and DECam data, respectively.

⁶ <https://github.com/pmvreeswijk/ZOGY>

⁷ <https://github.com/jah1994/PyTorchDIA>

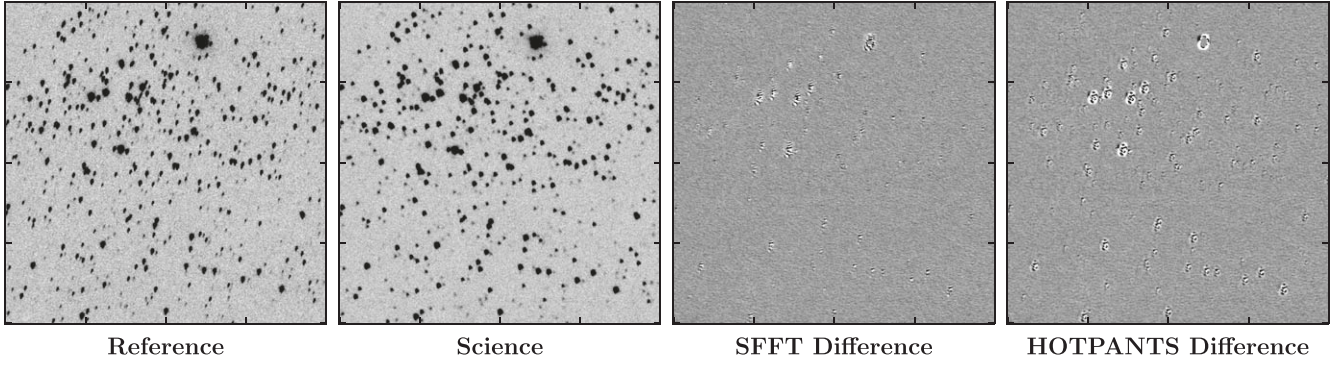


Figure 11. Performance of the image subtraction test using the TMTS images. The figure, from left to right, shows the input reference image *TMTS-REF*, science image *TMTS-SCI*, and the resulting difference images by SFFT and HOTPANTS.

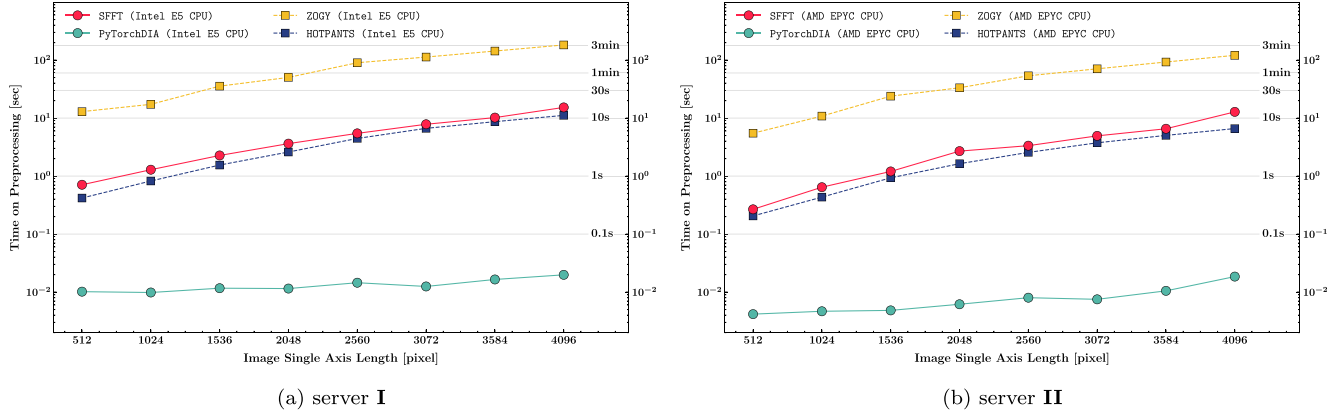


Figure 12. Computing time on preprocessing vs. the image size of the input data for different methods. The curves with different colors represent the measured time spent on pre-subtraction procedures using different methods on server I (left) and server II (right). Note that each test using SFFT, HOTPANTS, or PyTorchDIA has been repeated with a specified kernel size in each run. Given that the kernel size is irrelevant to preprocessing, we simply use the average time in the figure.

To explore the relationship between the computing time and input image size, we extract subimages of different sizes from the original TMTS image pair (i.e., *TMTS-REF* and *TMTS-SCI*). We then perform image subtractions for the trimmed image pairs on the two servers using the four tested methods. For SFFT, HOTPANTS, and PyTorchDIA, in order to test if the computational cost is sensitive to the size of the convolutional kernel, we run the image subtractions repeatedly with a specified kernel size from 15–25 pixels in each run (six runs in total).

Figure 12 shows the computing time on the preprocessing as a function of input image size. In general, larger input images consume more computing time on preprocessing. Among the tested methods, ZOGY requires the longest preprocessing time. For ZOGY, the preprocessing is composed mainly of fitting the PSF models for input images and calculating their flux ratio as well as the astrometric registration noises. The preprocessing time of SFFT and HOTPANTS are comparable and much shorter than that of ZOGY. For these two methods, the goal of the pre-subtraction procedures is to select an optimal set of sources for kernel determination. Like in Section 4.2, we have forced HOTPANTS to use the preselected set given by SFFT following Section 3.2. Note that SFFT further needs to perform an image masking based on the preselected set while HOTPANTS does not; that is why the preprocessing time of SFFT is slightly longer than that of HOTPANTS. For PyTorchDIA, the preprocessing only involves reading the input FITS images. Thus, it can achieve the minimal

preprocessing time. We note that the robust loss function used in PyTorchDIA is designed to be insensitive to the distraction sources, as a result, no source selection or image masking is involved in PyTorchDIA.

Figure 13 presents the computing time on the image subtraction versus the input image size. Unsurprisingly, larger images generally require more computing time for subtraction. The computing speeds for HOTPANTS and ZOGY are broadly comparable, with a typical cost of 40–90 s for 4K images. By contrast, the GPU-powered method PyTorchDIA brings a considerable speed-up. For the 4K images, the computing time is around 30 and 10 s on servers I and II, respectively. Note that PyTorchDIA only solves a spatially invariant kernel while other tested methods can accommodate spatially varying PSF. Remarkably, SFFT is almost an order of magnitude faster than PyTorchDIA. For the 4K images, the computing time is close to 4 and 1 s on servers I and II, respectively. As shown in Figure 13, the subtraction speed of SFFT is less affected by the kernel size compared to HOTPANTS and PyTorchDIA.

The second type of test aims to explore the computing expense on a set of images with diverse seeing conditions covering different sky areas. We randomly select 100 CCD images from the 23 DECam observations aliased by the initial DECam-OBS (see Table E1). Note that one DECam exposure produces 62 CCD images, each with a size of 2046×4094 pixels. As shown in Table E1, the DECam data were obtained under varying seeing conditions. Although the DECam exposures listed in Table E1 have similar telescope pointing,

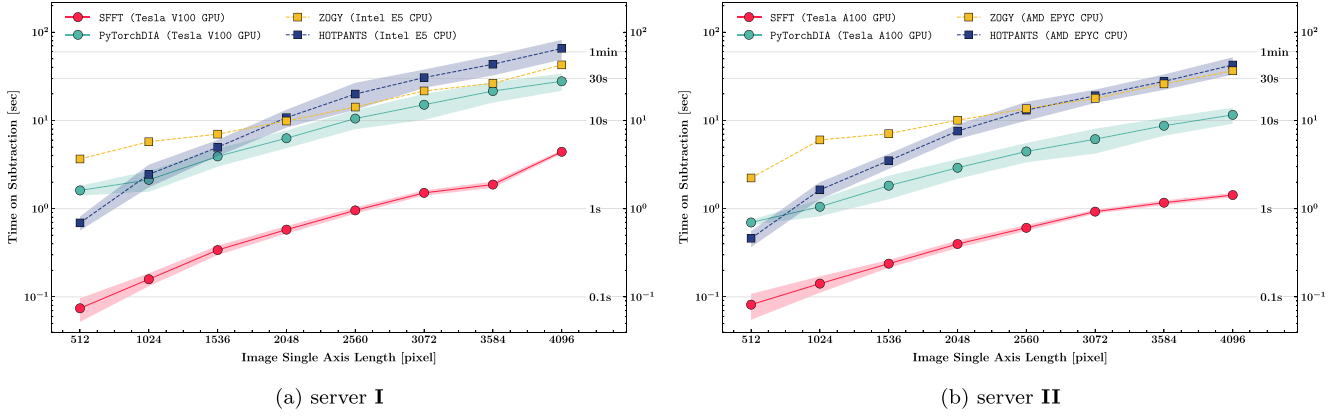


Figure 13. Computing time on image subtraction vs. the image size of input data for different methods. The curves with different colors represent the measured time spent on subtraction using different methods on server I (left) and server II (right). For SFFT, HOTPANTS, and PyTorchDIA, the data points in the curves represent the average time measured from the multiple runs, and the shaded areas indicate the corresponding 1σ standard deviation.

Table 2
Comparison of the Computing Time on Preprocessing and Subtraction Measured from the Tests on DECam Data Using Different Methods

Method	Time Cost of Preprocessing (s)		Time Cost of Subtraction (s)			
	Intel E5 CPU	AMD EPYC CPU	Intel E5 CPU	AMD EPY CPU	Tesla V100 GPU	Tesla A100 GPU
SFFT	5.75 ± 0.26	3.35 ± 0.15	...	...	1.61 ± 0.10	0.97 ± 0.06
PyTorchDIA	0.11 ± 0.01	0.16 ± 0.07	...	...	15.87 ± 3.25	6.45 ± 1.36
HOTPANTS	4.37 ± 0.85	2.43 ± 0.39	26.39 ± 8.50	15.39 ± 3.49	...	...
ZOGY	63.59 ± 6.42	42.28 ± 3.74	62.83 ± 9.16	41.01 ± 5.25	...	...

the arbitrarily selected CCD images can, to some extent, cover a variety of distributions of the observed stellar objects. In this test, the selected 100 images are regarded as science images, and we use the corresponding CCD images of *DECam-SREF* as their reference images. We perform image subtractions for the 100 image pairs on the two servers using the four tested methods. Unlike the above test, we do not specify a given kernel size but leave it to be automatically determined according to the seeing condition. Table 2 summarizes the time costs of preprocessing and image subtraction for the tested DECam data set. The results are broadly consistent with the first test. Again, SFFT shows a distinct advantage in the subtraction speed over the other tested methods. Given the image size, SFFT is the most stable method on subtraction time, with a percentage variation $\sim 6\%$ (the typical value is 20% for other methods).

We note that the software HOTPANTS also allows users to perform image subtraction without a preselected source list. In such an automatic mode, HOTPANTS will accomplish the source selection using its built-in functions. One may wonder if the automatic mode can be faster than the manual mode used in this paper. Hence, we additionally perform the image subtraction on the 100 image pairs using the automatic mode HOTPANTS. As shown in Table 3, the total computing time (preprocessing plus subtraction) for the two modes is very close with the automatic mode having a slightly higher variation.

For the crowded fields, the computational cost of SFFT will not rise to a level higher than the cost shown in this section. For preprocessing, crowded-flavor SFFT involves much simpler operations than sparse-flavor SFFT (see Section 3.1). On the other hand, signal-dominated input images will not prolong the computing time on image subtraction in Fourier space.

Table 3
Comparison of the Total Computing Time Measured from the Tests on DECam Data Using HOTPANTS in Two Modes

HOTPANTS Mode	Total Time Cost (s)	
	Intel E5	AMD EPYC
Manual (this work)	30.76 ± 8.81	17.82 ± 3.62
Automatic	30.75 ± 10.54	17.71 ± 4.40

6. Limitations and Future Works

To get a clear picture of the current SFFT and find its limitations, we summarize a variety of features from different perspectives for SFFT and other existing image subtraction methods, including PyTorchDIA, HOTPANTS, and ZOGY in Table 4. Although a general algorithm (e.g., Alard & Lupton 1998) can have multiple existing implementations, the scope of the comparisons in Table 4 is restricted: we have only considered a specific implementation for each algorithm. Some characteristics of a specific implementation may not necessarily be the intrinsic features of its underlying algorithm (see the note attached to Table 4).

Although SFFT uses the state-of-the-art DBFs as kernel basis functions, the current version does not contain an effective mechanism to address the overfitting problem of DBFs. In future work, we may try to incorporate existing solutions into SFFT; see, e.g., Becker et al. (2012), Bramich et al. (2016), and Masci et al. (2017). Moreover, unlike Alard & Lupton (1998), Bramich (2008), and Hitchcock et al. (2021), the kernel determination in SFFT is not weighted by pixel noise, which may introduce a possible bias toward the brighter pixels. It looks tricky to design a weighting scheme in Fourier space. Note that some practical implementations, including

Table 4
Characteristics of SFFT Compared with Other Existing Image Subtraction Methods

Method	Convolution Kernels			Accommodate Spatial Variations		
	Gaussian-function Basis	δ -function Basis	Regularized Kernel	Convolution Kernels	Photometric Scale Factor	Differential Background
SFFT	×	✓	×	✓	✓	✓
PyTorchDIA	×	✓	×	×	×	✓
HOTPANTS	✓	×	×	✓	×	✓
ZOGY	...	...	...	✓	✓	...
Method	Kernel Determination					
	Require PSF Knowledge	Require Isolated Objects	Rely on Rectangle Subareas	Weighting by Pixel Noise	Construct Least-squares Matrix	Symmetric to Image Exchange
SFFT	×	×	×	×	✓	×
PyTorchDIA	×	×	×	✓	×	×
HOTPANTS	×	✓	✓	×	✓	×
ZOGY	✓	✓	✓	...	×	✓
Method	Preprocessing before Subtraction			Computing Performance of Subtraction		
	Stamp Selection /Image Masking	PSF Modeling	Calculate Flux Ratio	Implemented for GPU	Built-in CPU Multithreading	Speed Sensitive to Kernel Size
SFFT	✓	×	×	✓	...	×
PyTorchDIA	×	×	×	✓	...	✓
HOTPANTS	✓	×	×	×	×	✓
ZOGY	...	✓	✓	×	✓	...

Note. For PyTorchDIA, the kernel regularization has been mentioned in Hitchcock et al. (2021) whereas no regularization has been implemented in the software PyTorchDIA; For HOTPANTS, the kernel determination relies on a set of stamps encompassing isolated objects. Note that requiring isolated objects is not a feature of the work in Alard & Lupton (1998). Moreover, pixel noise is not used in the matrix calculation in HOTPANTS though formulated in Alard & Lupton (1998); For ZOGY, all the convolutions involved in image subtraction are determined by precalculated PSFs. Generally, constructions of the PSF models are based on a sample of rectangle subimages (e.g., SExtractor VIGNETS) centered at isolated bright point sources.

HOTPANTS and Miller et al. (2008), also ignore the pixel weights in kernel determination.

As discussed in Zackay et al. (2016), the approach proposed by Alard & Lupton (1998) is not symmetric, i.e., changing the convolution direction of image subtraction will generally result in inconsistent difference images. Consequently, SFFT has some limitations inherited from the nature of this kind. In particular, when the direction matches an image to another one with better seeing, the deconvolution effect makes its way into the subtraction to amplify the noise and strengthen noise correlation on the difference image. In many cases, one can steer clear of the deconvolution effect by simply adjusting the direction. However, this strategy is not a panacea; e.g., it may be insufficient when the observations have elliptic-like PSF with varying orientations (Zackay et al. 2016).

Another limitation of the current SFFT is related to the computing cost of preprocessing. The image-masking scheme of SFFT introduced in Section 3 provides a safe and generic approach that has proved to be reliable through extensive tests by multiple transient surveys. However, its computing cost might become a potential hindrance for those projects that pursues extremely rapid transient detection. Fortunately, there is ample space for speed-up in a given survey program. One can modify the SFFT code to design a particular image-masking function for optimizing its overall computing expense. For example, the SExtractor catalogs used for the source selection can be provided by preceding modules (e.g., astrometric calibration) in the pipeline. When processing time-series data taken from the same pointing, one can skip the repeated photometry for the shared reference image to save

preprocessing time. One may also consider using a single predefined mask (fixed or seeing related) for all images in the time series. We note that PyTorchDIA can take very little time on preprocessing as it designs a specific loss function insensitive to outlying values, which provides an alternative to suppress the impact from distraction sources amid the kernel determination.

In the current implementations of SFFT, the subtraction has two flavors that enable image masking in different ways. However, there is no built-in discriminator for sparse fields and crowded fields in SFFT. One may specify the adopted flavor for each field to be processed. In future work, we may consider developing a more unified way to conduct image masking rather than dividing the observations into two subgroups. Besides, the limited amount of memory of GPU devices can constrain the applicability of the GPU-based SFFT to very large images. To allow for large image cases, we have provided a NumPy-based backend for SFFT without a GPU requirement. This is also helpful for users who do not have GPU resources available.

7. Summary and Conclusions

SFFT is a novel image subtraction algorithm formulated in Fourier space. Like in the classic framework of Alard & Lupton (1998), SFFT addresses the PSF discrepancy between two images by convolution with pixelized kernels that are decomposed into predefined basis functions. In our work, we adopted the δ -function basis proposed by Miller et al. (2008) for kernel flexibility and minimal hyper-parameters. SFFT

solves the least-squares problems of image subtraction in the Fourier domain, and it allows for space-varying PSF, photometric scaling, and sky background modeling across the entire field of view. In particular, SFFT can reduce the computationally intensive kernel determination to FFTs and element-wise matrix operations. By leveraging CUDA-enabled GPU acceleration, SFFT brings a great advancement in the computational performance of astronomical image subtraction with a speed-up of around one order of magnitude.

We demonstrated in this paper the power of our method on real astronomical images taken by a variety of telescopes. We show that SFFT can accommodate imaging data with diverse characteristics, including sparse fields, crowded fields, and under-sampled PSFs. The examples of SFFT for image subtraction are presented in Section 4. We confirm that SFFT yields a high quality of PSF matching when compared with the widely used image subtraction software HOTPANTS using a Gaussian kernel basis for models of the PSFs. A comprehensive investigation of the computing performance of SFFT is presented in Section 5. We find that SFFT is not only faster compared to other existing image subtraction methods but is also more stable and efficient.

We also explored the limitations of the current SFFT implementation in Section 6. One prominent weakness of the current SFFT is that the kernel solutions may suffer from the overfitting problem due to the high degrees of freedom of DBFs. A few approaches have been proposed to alleviate the overfitting issue of DBFs, e.g., regularizing the kernel solutions (Becker et al. 2012). However, the DBFs employed by the current SFFT are unregularized. The regularization techniques will be included in future works. In the meantime, proper image masking is required in the current version of SFFT to identify the pixels that are not correctly modeled by SFFT. In principle, SFFT also provides a generic and robust function to perform preprocessing of the data, which has been extensively tested with data from various transient surveys. In contrast to the high speed of the image subtraction, however, the computing performance of generic preprocessing is less remarkable. Given a particular time-domain program, we believe there is plenty of room for further optimization of the computing expense on the preprocessing.

We conclude that SFFT has the potential to be the optimal image subtraction engine for future time-domain surveys with massive data flows that require differential image subtractions for transient detection and precision differential photometry.

L.H. acknowledges the support from the National Natural Science Foundation of China (11761141001) and the Major Science and Technology Project of Qinghai Province (2019-ZJ-A10). L.W., X.C., and J.Y. acknowledge support from an NSF grant AST1817099. We thank Bo Yu and Tianrui Sun for useful discussions on the mathematical derivation and package developments. The authors are also grateful to an anonymous referee whose report has significantly improved this manuscript.

Software: Astropy (Astropy Collaboration et al. 2013), SciPy (Virtanen et al. 2020), Numpy (Harris et al. 2020), Matplotlib (Hunter 2007), CuPy (Okuta et al. 2017), PyCuda (Klöckner et al. 2012), ScikitCUDA (Givon et al. 2019), Numba (Lam et al. 2015), SExtractor (Bertin & Arnouts 1996), SWarp (Bertin 2010), PSFEx (Bertin 2011), HOTPANTS (Becker 2015), ZOGY (Zackay et al. 2016), PyTorchDIA (Hitchcock et al. 2021), SFFT (Hu et al. 2021).

Appendix A The Approximation in SFFT

Here we will prove the approximation applied to Equation (10). Given any image R and a kernel Q , where Q has a much smaller size than R as it is in most astronomical applications.

$$T^{\rho\epsilon}(R \otimes Q) \approx R^{\rho\epsilon} \otimes Q. \quad (A1)$$

Writing the left-hand side (LHS) and right-hand side (RHS) as follows:

$$\begin{aligned} \text{LHS}(r, s) &= r^{\rho\epsilon} s^{\epsilon} \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) R_p(r - c, s - d) \\ &= \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) [r^{\rho\epsilon} s^{\epsilon} R_p(r - c, s - d)]. \end{aligned} \quad (A2)$$

$$\begin{aligned} \text{RHS}(r, s) &= \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) R_p^{\rho\epsilon}(r - c, s - d) \\ &= \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) (T^{\rho\epsilon} R)_p(r - c, s - d). \end{aligned} \quad (A3)$$

Note that the indices of kernel Q are modified with the origin at its center pixel, and the subscript p is employed to indicate the periodic extension.

For r and s , $(r - c, s - d)$ is a vector within the image frame and $(r - c, s - d) \approx (r, s)$ for nearly all possible c and d .

$$\begin{aligned} \text{LHS}(r, s) &= \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) [r^{\rho\epsilon} s^{\epsilon} R(r - c, s - d)] \\ &\approx \sum_{c=-w_0}^{w'_0} \sum_{d=-w_1}^{w'_1} Q(c, d) [(r - c)^{\rho} (s - d)^{\epsilon} \\ &\quad R(r - c, s - d)] = \text{RHS}(r, s). \end{aligned} \quad (A4)$$

For the particular case $\rho = \epsilon = 0$, it becomes a rigorous equation rather than an approximation.

Appendix B Calculation Simplification

The δ -function basis facilitates further simplifications. Equation (19) will be re-expressed as

$$\begin{aligned} A_{\bar{i}\bar{j}\bar{\alpha}\bar{\beta}ij\alpha\beta} &= -\Omega_{\bar{i}\bar{j}ij}(\bar{\alpha}, \bar{\beta}) - \Omega_{\bar{i}\bar{j}ij}(-\alpha, -\beta) \\ &\quad + \Omega_{\bar{i}\bar{j}ij}(\bar{\alpha} - \alpha, \bar{\beta} - \beta) + \Omega_{\bar{i}\bar{j}ij}(0, 0) \\ &\quad \text{if } (\bar{\alpha}, \bar{\beta}) \neq \vec{0} \text{ and } (\alpha, \beta) \neq \vec{0} \\ A_{\bar{i}\bar{j}\bar{\alpha}\bar{\beta}ij\alpha\beta} &= \Omega_{\bar{i}\bar{j}ij}(-\alpha, -\beta) - \Omega_{\bar{i}\bar{j}ij}(0, 0) \\ &\quad \text{if } (\bar{\alpha}, \bar{\beta}) = \vec{0} \text{ and } (\alpha, \beta) \neq \vec{0} \\ A_{\bar{i}\bar{j}\bar{\alpha}\bar{\beta}ij\alpha\beta} &= \Omega_{\bar{i}\bar{j}ij}(\bar{\alpha}, \bar{\beta}) - \Omega_{\bar{i}\bar{j}ij}(0, 0) \\ &\quad \text{if } (\bar{\alpha}, \bar{\beta}) \neq \vec{0} \text{ and } (\alpha, \beta) = \vec{0} \\ A_{\bar{i}\bar{j}\bar{\alpha}\bar{\beta}ij\alpha\beta} &= \Omega_{\bar{i}\bar{j}ij}(0, 0) \\ &\quad \text{if } (\bar{\alpha}, \bar{\beta}) = \vec{0} \text{ and } (\alpha, \beta) = \vec{0}, \end{aligned} \quad (B1)$$

where

$$\Omega_{\bar{i}\bar{j}}(\rho, \epsilon) = \hat{\Omega}_{\bar{i}\bar{j}}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Omega}_{\bar{i}\bar{j}} = \frac{1}{N_0 N_1} \Re [\text{DFT}(\widehat{R^{\bar{i}\bar{j}}}(\widehat{R^{\bar{i}\bar{j}}})^*)]. \quad (\text{B2})$$

Equation (20) can be rewritten as

$$B_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}pq} = \Lambda_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\bar{\alpha}, \bar{\beta}) - \Lambda_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\bar{\alpha}, \bar{\beta}) \neq \vec{0} \\ B_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}pq} = \Lambda_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\bar{\alpha}, \bar{\beta}) = \vec{0}, \quad (\text{B3})$$

where

$$\Lambda_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\rho, \epsilon) = \hat{\Lambda}_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Lambda}_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}} = \Re [\text{DFT}(\widehat{R^{\bar{i}\bar{j}}}(\widehat{T^{\bar{p}q}})^*)]. \quad (\text{B4})$$

Equation (21) is written as

$$\tilde{B}_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}} = \Psi_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}}(-\alpha, -\beta) - \Psi_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\alpha, \beta) \neq \vec{0} \\ \tilde{B}_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}} = \Psi_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\alpha, \beta) = \vec{0}, \quad (\text{B5})$$

where

$$\Psi_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}}(\rho, \epsilon) = \hat{\Psi}_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Psi}_{\bar{p}\bar{q}}^{\bar{\alpha}\bar{\beta}} = \Re [\text{DFT}(\widehat{T^{\bar{p}\bar{q}}}(\widehat{R^{\bar{i}\bar{j}}})^*)]. \quad (\text{B6})$$

Equation (22) turns out to be

$$C_{\bar{p}\bar{q}pq} = \Phi_{\bar{p}\bar{q}pq}(0, 0), \quad (\text{B7})$$

where

$$\Phi_{\bar{p}\bar{q}pq}(\rho, \epsilon) = \hat{\Phi}_{\bar{p}\bar{q}pq}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Phi}_{\bar{p}\bar{q}pq} = N_0 N_1 \Re [\text{DFT}(\widehat{T^{\bar{p}\bar{q}}}(\widehat{T^{\bar{p}q}})^*)] \quad (\text{B8})$$

Likewise, Equation (23) becomes

$$D_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}} = \Theta_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\bar{\alpha}, \bar{\beta}) - \Theta_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\bar{\alpha}, \bar{\beta}) \neq \vec{0} \\ D_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}} = \Theta_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(0, 0) \text{ if } (\bar{\alpha}, \bar{\beta}) = \vec{0}, \quad (\text{B9})$$

where

$$\Theta_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\rho, \epsilon) = \hat{\Theta}_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Theta}_{\bar{i}\bar{j}}^{\bar{\alpha}\bar{\beta}} = \Re [\text{DFT}(\widehat{S^*}(\widehat{I^{\bar{i}\bar{j}}})^*)]. \quad (\text{B10})$$

Equation (24) on the RHS becomes

$$E_{\bar{p}\bar{q}} = \Delta_{\bar{p}\bar{q}}(0, 0), \quad (\text{B11})$$

where

$$\Delta_{\bar{p}\bar{q}}(\rho, \epsilon) = \hat{\Delta}_{\bar{p}\bar{q}}(\rho \bmod N_0, \epsilon \bmod N_1) \\ \hat{\Delta}_{\bar{p}\bar{q}} = N_0 N_1 \Re [\widehat{S^*} \widehat{T^{\bar{p}\bar{q}}}]. \quad (\text{B12})$$

Now Equations (19)–(24) have been converted to Equations (B1), (B3), (B5), (B7), (B9), and (B11). These equations used to establish the linear system are mainly comprised of DFTs and element-wise matrix operations, which allows for highly efficient parallel computations on a GPU architecture.

Appendix C Noise Decorrelation

Like in other subtraction methods emanated from Alard & Lupton (1998), SFFT will introduce correlated errors on the difference image through the convolution process. For the cross-convolution approach, Zackay et al. (2016) have proved that one can whiten the difference image by one convolution with a decorrelation kernel. Later, Reiss & Lupton (2016) presented a noise decorrelation strategy for image subtraction under the framework of Alard & Lupton (1998). Here we extend this to a more general case. That is, the subtraction can be performed with images constructed from the average or median co-addition of individual observations, and the original images are preprocessed by convolutions to match their PSFs before the co-additions.

The following derivation does not consider the spatial variation of convolution kernels across the field of view. However, it is appropriate to perform the operation by splitting the images into smaller segments to minimize the effect of PSF variation on correlated background noise. Likewise, the spatial variation of the background map for the input images is not considered. Note that a constant map has nothing to do with the noise correlation. Therefore, it is appropriate to assume that all the input images have zero mean background in our derivation. Note that noise correlation caused by resampling for image alignment and drizzling is not considered here, as such operations cannot be simplified as convolution, and it is not easy to formulate how these procedures affect the statistical properties of the noise. Finally, we assume that the noise of the original images before convolution to be uncorrelated Poisson noise.

Given M science images $S_j \sim \mathcal{N}(0, \sigma_{S_j}^2)$ with j being integers from 0, $M-1$, and reference images $R_i \sim \mathcal{N}(0, \sigma_{R_i}^2)$ with i being integers from 0, $N-1$, respectively. We consider the generic case of subsequent average or median co-additions to produce the stacked science and reference images. The image subtraction employed to obtain the final difference image D is given by

$$D = \sum_{j=0}^{M-1} S_j \otimes G_j - [\sum_{i=0}^{N-1} R_i \otimes H_i] \otimes K. \quad (\text{C1})$$

For simplicity, the convolution kernels G_j and H_i for PSF homogenization are divided by a number of science (S_j) and reference (R_i) images M and N , respectively. Assuming that a decorrelation kernel Q exists that can whiten the noise of D , we can calculate Q using the equation $\mathcal{D} = D \otimes Q$, where $\mathcal{D} \sim \mathcal{N}(0, \sigma_{\mathcal{D}}^2)$.

The covariance matrix of $\mathcal{D}$ in Fourier space can be written as

$$\text{cov}(\widehat{\mathcal{D}}(p), \widehat{\mathcal{D}}(q)) = L_x L_y \delta(p, q) \widehat{Q}(p) \widehat{Q}(q)^* \\ \times \left\{ \sum_j \sigma_{S_j}^2 \widehat{G}_j(p) \widehat{G}_j(q)^* \right. \\ \left. + \sum_i \sigma_{R_i}^2 \widehat{K}(p) \widehat{K}(q)^* \widehat{H}_i(p) \widehat{H}_i(q)^* \right\}, \quad (\text{C2})$$

where L_x and L_y are the image dimensions in the x and y directions, respectively, and p and q are the pixel indices of the image frame.

The Fourier transform of the decorrelation kernel Q can be derived by setting $q = p$,

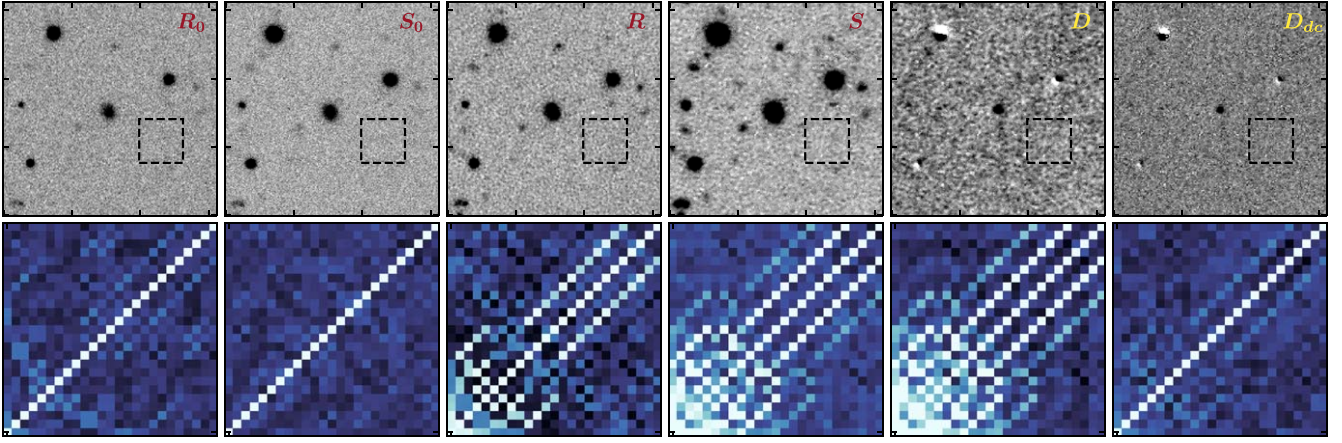


Figure 14. Noise decorrelation test with DECam data. The upper panels show the zoomed-in region of SN candidate AT 2018fjd (R.A. = 21:12:25.980 decl. = $-66:00:56.50$) discovered by DECamERON (Mould et al. 2017; Hu et al. 2018). R_0 (S_0) is from individual observations *DECam-OBS18a* (*DECam-OBS04a*). R (S) is the stacked reference (science) image. The subtraction between R and S results in the difference image D , and D_{dc} is the decorrelated difference image. The black dashed boxes (41×41 pixels) are selected on the background region in the vicinity of AT 2018fjd. The lower panel shows the corresponding covariance matrix measured from the boxes. To probe the local correlation of background pixels, we constructed a multivariate random variable X for the flux of adjacent pixels with fixed positional relationship, where $X = (X_{(0,0)}, X_{(1,0)}, X_{(-1,0)}, X_{(0,1)}, X_{(0,-1)}, X_{(1,1)}, X_{(-1,1)}, X_{(-1,-1)}, X_{(2,0)}, X_{(-2,0)}, X_{(0,2)}, X_{(0,-2)}, X_{(3,0)}, X_{(-3,0)}, X_{(0,3)}, X_{(0,-3)}, X_{(4,0)}, X_{(-4,0)}, X_{(0,4)}, X_{(0,-4)}, X_{(5,0)}, X_{(-5,0)}, X_{(0,5)}, X_{(0,-5)})$. Note the subscripts represent the relative pixel locations with respect to the first element. Only the pixels within the boxes are allowed to be involved in the random vector. The covariance matrix is calculated by exhausting all possible realizations of X .

$$|\widehat{Q}(p)|^2 = \sigma_{\mathcal{D}}^2 / \left\{ \sum_j \sigma_{S_j}^2 |\widehat{G}_j(p)|^2 + \sum_i \sigma_{R_i}^2 |\widehat{K}(p)|^2 |\widehat{H}_i(p)|^2 \right\}. \quad (\text{C3})$$

The final solution to $\widehat{Q}$ should be conjugate symmetric to ensure its inverse Fourier transformation to be real. Here we consider the most straightforward solution with $\widehat{Q}$ being a real function only, that is

$$\widehat{Q}_0 = \sqrt{\sigma_{\mathcal{D}}^2 / \{ \sum_j \sigma_{S_j}^2 |\widehat{G}_j|^2 + \sum_i \sigma_{R_i}^2 |\widehat{K}|^2 |\widehat{H}_i|^2 \}}. \quad (\text{C4})$$

Note $\sigma_{\mathcal{D}}$ is the desired noise of the decorrelated difference image. In principle, it should be set to be as close as possible to the true background in regions free of astronomical objects. In practice, this normalization can also be set by requiring the decorrelation procedure to preserve the flux zero-point of the images. In the latter case, SFFT simply normalizes the decorrelation kernel Q to have a unit kernel sum.

There is a major difference between SFFT and **ZOGY** for noise decorrelation, although they share the same statistical principles. In **ZOGY**, convolution kernels for co-addition and subtraction are tied to the PSFs of the images. SFFT does not need to model the PSFs for the images S_j and R_i for the noise decorrelation process. Only the match kernels G_j , H_i , and K are needed for SFFT. This can be even more convenient for images with simple co-additions of original observations without matching the PSFs, such as observations taken under stable seeing conditions, only the matching kernel K is needed for SFFT, and no PSF construction is necessary.

We select 10 DECam images from Table E1 to test the performance of noise decorrelation. The generic case of image subtraction follows Equation (C1), with $\{R_0, R_1, R_2, R_3, R_4\} = \{\textit{DECam-OBS18a}, \textit{DECam-OBS18b}, \textit{DECam-OBS18c}, \textit{DECam-OBS18d}, \textit{DECam-OBS18e}\}$ being the individual reference images, and $\{S_0, S_1, S_2, S_3, S_4\} = \{\textit{DECam-OBS04a}, \textit{DECam-OBS04b}, \textit{DECam-OBS04c}, \textit{DECam-OBS04d}, \textit{DECam-OBS04e}\}$ being the individual science images. All the images have been astronomically aligned to S_0 by **SWarp**. We create the stacked reference image by median co-addition of the five individual reference images with PSF-homogenization matching to R_0 . Similarly, the stacked science image is the median coadd of the five individual science images after matching their PSFs to S_0 . The final subtraction is performed between the stacked reference and stacked science image. Noise decorrelation for the difference image is conducted by convolving the decorrelation kernel calculated by Equation (C4). As the spatial variation has been ignored, here, the example only focuses on a small area around the supernova candidate AT 2018fjd discovered in the DECamERON survey (Hu et al. 2018).

We trace the background pixel correlation along with the coadd and subtraction processes, as shown in Figure 14. Note that image alignment can also introduce pixel correlation via a localized resampling function (LANCZOS3 function of **SWarp** in this case). One can notice that the pixel correlation in the resampled image R_0 is already slightly higher than the un-resampled image S_0 . The correlation is increased drastically in coadded images as multiple convolutions for PSF homogenization come into play in this process. It becomes even stronger after the final subtraction. Nevertheless, our decorrelation scenario can effectively reduce the noise correlation on the final difference image to a quite low level comparable with that of a resampled individual image R_0 .

Appendix D

Software Parameters

Here we show the default parameters of the softwares used in this paper in Tables D1–D4.

Table D1
SFFT Parameters

Parameter	Value	Note	Comment
BeltHW	0.2	Half-width of the point-source belt	Only for sparse flavor
MAGD_THRESH	0.12	Outlier threshold to reject variable sources (mag)	Only for sparse flavor
KerPolyOrder	2	Spatial order of kernel variation	...
BGPolyOrder	2	Spatial order of background variation	...
KerHWRatio	2	The ratio between seeing and convolution kernel half-width	...
ConstPhotRatio	True	Constant photometric ratio between reference and science	...

Table D2
HOTPANTS Parameters

Parameter	Value	Note
r	$2 \times \text{FWHM}_L$	Convolution kernel half-width (pixel)
rss	$6 \times \text{FWHM}_L$	Sub-stamp half-width (pixel)
nsx	$NX/200$	Image segmentation of stamps (x -axis)
nsy	$NY/200$	Image segmentation of stamps (y -axis)
ko	2	Spatial order of kernel variation
bgo	2	Spatial order of background variation
tu	Saturation recorded in reference header	Upper valid flux of the reference image
tl	$\text{sky}(\text{reference}) - 10 \times \text{skysigma}(\text{reference})$	Lower valid flux of the reference image
iu	Saturation recorded in science header	Upper valid flux of the science image
il	$\text{sky}(\text{science}) - 10 \times \text{skysigma}(\text{science})$	Lower valid flux of the science image

Note. FWHM_L is the worst seeing measured from the input image pair. Note the ratio of 2 in parameter r is consistent with the default configuration of SFFT (see KerHWRatio). The parameters ko and bgo are equivalent to SFFT parameters KerPolyOrder and BGPolyOrder, respectively. The symbols NX and NY indicate the image size along the x -axis and y -axis, respectively. Sky and skysigma describe the sky background measured by the MMM algorithm. The HOTPANTS configurations here largely refer to those in the iPTF image subtraction pipeline (Cao et al. 2016).

Table D3
ZOGY Parameters

Parameter	Value	Note
subimage_size	256	Size of subimages (pixel)
subimage_border	32	Border around subimage to avoid edge effects (pixel)
fratio_local	F	Flux ratio from subimage (T) or full frame (F)
psf_poldeg	2	Polynomial degree for PSF spatial variation
size_vignet	49	Size of the SExtractor VIGNETS for PSF construction
nthreads	32	Number for CPU multithreading

Table D4
PyTorchDIA Parameters

Parameter	Value	Note
loss_fn	robust	Loss function (robust or Gaussian)
flat	1	Initialization value of the convolution kernel
ks	$4 \times \text{FWHM}_L + 1$	Kernel size
poly_degree	2	Degree of the polynomial for the background fit

Note. FWHM_L is the worst seeing measured from the input image pair.

Table E1
Specifications of the Test Data

Instrument	Alias File	File Name	Exposure Time (s)	FWHM (arcsec)	5 σ Lim. Mag (mag)
ZTF	<i>ZTF-REF</i>	ztf_001735_zg_c01_q2_refimg	1200	2.02	22.89
ZTF	<i>ZTF-SCI</i>	ztf_20180705481609_001735_zg_c01_o_q2_sciimg	30	1.80	20.68
AST3-II	<i>AST-REF</i>	a0331.94	60	2.41	...
AST3-II	<i>AST-SCI</i>	a0405.162	60	2.96	...
TESS	<i>TESS-REF</i>	tess2018258205941-s0002-1-1-0121-s_ffic.001	1440	36.95	...
TESS	<i>TESS-SCI</i>	tess2018258225941-s0002-1-1-0121-s_ffic.001	1440	36.89	...
DECam	<i>DECam-SREF</i>	c4d_180806_052738_ooi_i_v1	330	0.85	24.35
DECam	<i>DECam-PTAR</i>	c4d_150821_001224_ooi_i_v1	330	1.40	22.73
DECam	<i>DECam-OBS18a</i>	c4d_160818_043331_ooi_i_v1	330	1.04	23.28
DECam	<i>DECam-OBS18b</i>	c4d_160818_020339_ooi_i_v1	330	1.07	23.35
DECam	<i>DECam-OBS18c</i>	c4d_160818_030325_ooi_i_v1	330	0.96	23.45
DECam	<i>DECam-OBS18d</i>	c4d_160818_053313_ooi_i_v1	330	1.12	23.09
DECam	<i>DECam-OBS18e</i>	c4d_160818_065148_ooi_i_v1	330	1.01	23.32
DECam	<i>DECam-OBS02a</i>	c4d_180802_052807_ooi_i_v1	330	1.38	23.40
DECam	<i>DECam-OBS02b</i>	c4d_180802_055800_ooi_i_v1	330	1.31	23.55
DECam	<i>DECam-OBS02c</i>	c4d_180802_062754_ooi_i_v1	330	1.22	23.60
DECam	<i>DECam-OBS02d</i>	c4d_180802_065746_ooi_i_v1	330	1.41	23.33
DECam	<i>DECam-OBS02e</i>	c4d_180802_072740_ooi_i_v1	330	1.21	23.58
DECam	<i>DECam-OBS02f</i>	c4d_180802_075732_ooi_i_v1	330	1.29	23.62
DECam	<i>DECam-OBS02g</i>	c4d_180802_085719_ooi_i_v1	330	1.24	23.55
DECam	<i>DECam-OBS02h</i>	c4d_180802_092739_ooi_i_v1	330	1.35	23.42
DECam	<i>DECam-OBS04a</i>	c4d_180804_092331_ooi_i_v1	330	1.41	23.50
DECam	<i>DECam-OBS04b</i>	c4d_180804_052409_ooi_i_v1	330	1.37	23.62
DECam	<i>DECam-OBS04c</i>	c4d_180804_055402_ooi_i_v1	330	1.57	23.43
DECam	<i>DECam-OBS04d</i>	c4d_180804_075344_ooi_i_v1	330	1.36	23.61
DECam	<i>DECam-OBS04e</i>	c4d_180804_082338_ooi_i_v1	330	1.10	23.89
DECam	<i>DECam-OBS04f</i>	c4d_180804_045412_ooi_i_v1	330	1.46	23.45
DECam	<i>DECam-OBS04g</i>	c4d_180804_062356_ooi_i_v1	330	1.68	23.35
DECam	<i>DECam-OBS04h</i>	c4d_180804_065350_ooi_i_v1	330	1.45	23.58
DECam	<i>DECam-OBS04i</i>	c4d_180804_072342_ooi_i_v1	330	1.66	23.22
DECam	<i>DECam-OBS04j</i>	c4d_180804_085332_ooi_i_v1	330	1.40	23.51
TMTS	<i>TMTS-REF</i>	f20191103_1_NT0023_L_1965_1970	60	6.45	...
TMTS	<i>TMTS-SCI</i>	f20191028_1_NT0023_L_755_760	60	7.46	...

Note. *ZTF-REF*, directly retrieved from ZTF Data Release 3, is coadded from 40 individual observations as a deep reference. *TESS-REF* and *TESS-SCI* are FFIs recorded at a 30 minute cadence, with an effective integration time of 1440 s. *TMTS-REF* (*TMTS-SCI*) is the median stacked from six consecutive observations, each of which has an exposure time of 10 s. Other images in the table are single-exposure observations. Each DECam exposure is comprised of 62 CCD images, the FWHM, and the limiting magnitude for each exposure in the table shows the median measurement over all CCD tiles.

Appendix E Specifications of the Test Data

Here we show the specifications of the test data used in this paper in Table E1.

ORCID iDs

Lei Hu  <https://orcid.org/0000-0001-7201-1938>
 Lifan Wang  <https://orcid.org/0000-0001-7092-9374>
 Xingzhao Chen  <https://orcid.org/0000-0003-3021-4897>
 Jiawen Yang  <https://orcid.org/0000-0002-1376-0987>

References

- Akhlaghi, M., & Ichikawa, T. 2015, *ApJS*, **220**, 1
- Alard, C. 2000, *A&AS*, **144**, 363
- Alard, C., & Lupton, R. H. 1998, *ApJ*, **503**, 325
- Albrow, M. D., Horne, K., Bramich, D. M., et al. 2009, *MNRAS*, **397**, 2099
- Astropy Collaboration, Robitaille, T. P., & Tollerud, E. J. 2013, *A&A*, **558**, A33
- Baillon, P., Bouquet, A., Giraud-Heraud, Y., & Kaplan, J. 1993, *A&A*, **277**, 1
- Becker, A. 2015, HOTPANTS: High Order Transform of PSF ANd Template Subtraction, Astrophysics Source Code Library, ascl:[1504.004](https://ui.adsabs.org/abs/2015ascl..1504A004)
- Becker, A. C., Homrighausen, D., Connolly, A. J., et al. 2012, *MNRAS*, **425**, 1341
- Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, *PASP*, **131**, 018002
- Bertin, E. 2010, SWarp: Resampling and Co-adding FITS Images Together, Astrophysics Source Code Library, ascl:[1010.068](https://ui.adsabs.org/abs/2010ascl..1010B08)
- Bertin, E. 2011, in ASP Conf. Ser. 442, Astronomical Data Analysis Software and Systems XX, ed. I. N. Evans et al. (San Francisco, CA: ASP), 435
- Bertin, E., & Arnouts, S. 1996, *A&AS*, **117**, 393
- Blum, R. D., Burleigh, K., Dey, A., et al. 2016, AAS Meeting, **228**, 317.01
- Bramich, D. M. 2008, *MNRAS*, **386**, L77
- Bramich, D. M., Horne, K., Albrow, M. D., et al. 2013, *MNRAS*, **428**, 2275
- Bramich, D. M., Horne, K., Alsubai, K. A., et al. 2016, *MNRAS*, **457**, 542
- Cao, Y., Nugent, P. E., & Kasliwal, M. M. 2016, *PASP*, **128**, 114502
- Crotts, A. P. S. 1992, *ApJL*, **399**, L43
- Gal-Yam, A., Maoz, D., Guhathakurta, P., & Filippenko, A. V. 2008, *ApJ*, **680**, 550
- Givon, L. E., Unterthiner, T., Erichson, N. B., et al. 2019, scikit-cuda 0.5.3: a Python interface to GPU-powered 1328 libraries, v0.5.3, Zenodo, doi:[10.5281/zenodo.3229433](https://doi.org/10.5281/zenodo.3229433)
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
- Hartung, S., Shukla, H., Miller, J. P., & Pennypacker, C. 2012, in 19th IEEE Int. Conf. on Image Processing (Piscataway, NJ: IEEE), 1685
- Hitchcock, J. A., Hundertmark, M., Foreman-Mackey, D., et al. 2021, *MNRAS*, **504**, 3561
- Honscheid, K., & DePoy, D. L. 2008, arXiv:[0810.3600](https://arxiv.org/abs/0810.3600)

- Hough, P. 1959, Conf. Proc. C, 590914, 554, <https://inspirehep.net/literature/919922>
- Hu, L., Wang, L., & Chen, X. 2021, SFFT v1.0.3, Zenodo, doi:[10.5281/zenodo.5521622](https://doi.org/10.5281/zenodo.5521622)
- Hu, L., Wang, L., Koekemoer, A., et al. 2018, Transient Name Server Discovery Report, [2018-1238](https://arxiv.org/abs/2018-1238)
- Hunter, J. D. 2007, CSE, **9**, 90
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, [ApJ](https://arxiv.org/abs/1901.09999), **873**, 111
- Klöckner, A., Pinto, N., Lee, Y., et al. 2012, [ParC](https://arxiv.org/abs/1205.3815), **38**, 157
- Kochanski, G. P., Tyson, J. A., & Fischer, P. 1996, [AJ](https://arxiv.org/abs/1901.09999), **111**, 1444
- Lam, S. K., Pitrou, A., & Seibert, S. 2015, in Proc. Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15 (New York: ACM)
- Mao, S., & Paczynski, B. 1991, [ApJL](https://arxiv.org/abs/1901.09999), **374**, L37
- Masci, F. J., Laher, R. R., Rebbapragada, U. D., et al. 2017, [PASP](https://arxiv.org/abs/1901.09999), **129**, 014002
- Miller, J. P., Pennypacker, C. R., & White, G. L. 2008, [PASP](https://arxiv.org/abs/1901.09999), **120**, 449
- Morganson, E., Gruendl, R. A., Menanteau, F., et al. 2018, [PASP](https://arxiv.org/abs/1901.09999), **130**, 074501
- Mould, J., Abbott, T., Cooke, J., et al. 2017, [SciBu](https://arxiv.org/abs/1901.09999), **62**, 675
- Okuta, R., Unno, Y., Nishino, D., Hido, S., & Loomis, C. 2017, in 31st Conf. on Neural Information Processing Systems (NIPS 2017), http://learningsys.org/nips17/assets/papers/paper_16.pdf
- Phillips, A. C., & Davis, L. E. 1995, in ASP Conf. Ser. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco, CA: ASP), 297
- Price, P. A., & Magnier, E. A. 2019, arXiv:[1901.09999](https://arxiv.org/abs/1901.09999)
- Reiss, D. J., & Lupton, R. H. 2016, DMTN-021: 1386 Implementation of Image Difference Decorrelation, 1387, Zenodo, doi:[10.5281/zenodo.192833](https://doi.org/10.5281/zenodo.192833)
- Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2015, [JATIS](https://arxiv.org/abs/1901.09999), **1**, 014003
- Sedaghat, N., & Mahabal, A. 2018, [MNRAS](https://arxiv.org/abs/1901.09999), **476**, 5365
- Sumi, T., Abe, F., Bond, I. A., et al. 2003, [ApJ](https://arxiv.org/abs/1901.09999), **591**, 204
- Sumi, T., Bennett, D. P., Bond, I. A., et al. 2013, [ApJ](https://arxiv.org/abs/1901.09999), **778**, 150
- Sumi, T., Woźniak, P. R., Udalski, A., et al. 2006, [ApJ](https://arxiv.org/abs/1901.09999), **636**, 240
- Tomaney, A. B., & Crofts, A. P. S. 1996, [AJ](https://arxiv.org/abs/1901.09999), **112**, 2872
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, [NatMe](https://arxiv.org/abs/1901.09999), **17**, 261
- Wang, L., Ma, B., Li, G., et al. 2017, [AJ](https://arxiv.org/abs/1901.09999), **153**, 104
- Watson, C. L., Henden, A. A., & Price, A. 2006, SASS, **25**, 47
- Yuan, F., & Akerlof, C. W. 2008, [ApJ](https://arxiv.org/abs/1901.09999), **677**, 808
- Yuan, X., Cui, X., Gu, B., et al. 2014, [Proc. SPIE](https://arxiv.org/abs/1901.09999), **9145**, 91450F
- Zackay, B., Ofek, E. O., & Gal-Yam, A. 2016, [ApJ](https://arxiv.org/abs/1901.09999), **830**, 27
- Zhang, J.-C., Wang, X.-F., Mo, J., et al. 2020, [PASP](https://arxiv.org/abs/1901.09999), **132**, 125001
- Zwicky, F. 1964, AnAp, **27**, 300