Joint survey processing: combined resampling and convolution for galaxy modelling and deblending

Rény Josepha, Peter Melchiorb, Fred Moolekampc

aDepartment of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA
bCenter for Statistics & Machine Learning, Princeton University, Princeton, NJ 08544, USA
cSchool of Physics and Astronomy, Rochester Institute of Technology, Rochester, NY 14623, USA

Abstract

We present an extension of the multi-band galaxy fitting method scarlet which allows the joint modeling of astronomical images from different instruments, by performing simultaneous resampling and convolution. We introduce a fast and formally accurate linear projection operation that maps a pixelated model at a given resolution onto an observation frame with a different point spread function and pixel scale. We test our implementation against the well-tested resampling and convolution method in galsim on simulated images mimicking observations with the Euclid space telescope and the Vera C. Rubin Observatory, and find that it reduces interpolation errors by an order of magnitude or more compared to galsim default settings. Tests with a wide range of levels of blending show more accurate galaxy models from joint modeling of Euclid and Rubin images compared to separate modeling of each survey by up to an order of magnitude. Our results demonstrate, for the first time, the feasibility and utility of performing non-parametric pixel-level data fusion of overlapping imaging surveys. All results can be reproduced with the specific versions of the codes and notebooks used in this study.

Keywords: surveys, resampling, multi-band imaging, deblending, multi-resolution, data fusion, image processing

1. Introduction

Every large-scale space or ground-based survey leads to a wide variety of data products, each one with characteristics unique to the instrument and survey. For instance, the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST) (Ivezić et al., 2019) will provide multiband observations down to r-band magnitude 27.5 over 18,000 deg$^2$. The Nancy Grace Roman Space Telescope (Spergel et al., 2015; Akeson et al., 2019) will produce high resolution images in seven bands in the visible and near-infrared range over 2,000 deg$^2$. The Euclid mission (Laureijs et al., 2011) will produce wide-band high resolution imaging over 15,000 deg$^2$ also in the optical and near-infrared range. While each of these surveys are expected to yield unprecedented scientific results for the astronomical and cosmology community, we should also recognize that they are highly complementary. What the Euclid survey lacks in sensitivity and spectral resolution in the visible range, LSST can compensate with deep multi-band observations. Conversely, what LSST’s images, taken with the Simonyi Survey Telescope, lack in spatial resolution, the two space telescopes Roman and Euclid can provide images with ~ 2 times higher pixel resolution.

Several studies have addressed the complementary nature of these surveys (Capak et al., 2019; Eifler et al., 2019; Rhodes et al., 2017, 2019; Scolnic et al., 2018; Chary et al., 2019, 2020), including estimated gains from combining observations at the catalog or at the pixel-level, for specific science cases, such as galaxy shape measurement (Schulmann et al., 2019), weak lensing and galaxy clustering (Eifler et al., 2020), photometric redshift estimates (Graham et al., 2020), observations of tidal stellar streams (Laine et al., 2018), exoplanet searches (Bachelet & Penny, 2019; Street et al., 2018) and solar system object tracking (Snodgrass et al., 2018).

To jointly model observations at different resolutions at the pixel level, we could choose to employ parameterized models, which potentially renders changes of resolution or convolution with the point spread function (PSF) to be analytically expressible. But parameterized models do not fully reflect the variety of galaxy morphologies. While that can be acceptable for ground-based surveys, it becomes a noticeable shortcoming for observations from space, where intricate structures are visible in most galaxies. For non-parametric methods, we need to build a resampling model that accounts for the difference in pixel size and positions between observations, the difference between the telescopes PSFs and the instruments’ pixel integrations. The problem of resampling is well studied in the astronomical literature, and a number of methods exist, e.g. for forced photometry (Merlin et al., 2016), for upsampling of undersampled images (Bertin, 2011; Ngol ‘e Mboula et al., 2015; Rowe et al., 2011), and general-purpose image resampling (Bernstein & Gruen, 2014, BG hereafter).

In this paper we implement a resampling scheme with minimal assumptions and approximations. We then integrate it into the galaxy modeling framework Oscarlet$^1$ (Melchior et al., 2018), which specialises in multi-band modeling and deblend-

---

1https://github.com/pmelchior/scarlet
ing, i.e., separating individual astrophysical objects overlapping on the plane of the sky. We carefully design our algorithm to perform a linear resampling that explicitly accounts for the instrumental and optical response of the image acquisition process. This allows the scarlet package to explicitly account for the resampling operator in the optimization process of modeling images.

An alternative for pixel-level joint modeling and deblending at two different resolutions has been recently published in Arcelin et al. (2020). In that work, the authors use a variational autoencoder that learns to recover the morphologies of galaxies by training them on blended images from multiple instruments. In contrast, we implement an explicit resampling operation that transforms a high-resolution image into a lower-resolution image to match any observation. By implementing this operation separately, can use with any type of parameterization for the high-resolution model and any type of inference.

In this paper, we restrict ourselves to presenting the resampling method and validating that pixel-level joint survey deblending leads to better reconstruction of light profiles than single-resolution deblending. A more quantitative study of the gains from joint survey processing requires using metrics that are specific to the individual science cases of interest. This is outside of the scope of this paper. A follow-up publication will study the gains from using multiple resolution images specifically for object detection and its impact on deblending.

The paper is organised as follows: we introduce our resampling approach in Section 2. Section 3 discusses our implementation of a practical and faster resampling strategy in the scarlet framework. We compare the runtime and accuracy of the resampling with the reference BG method implemented in 

GalSim
two (Rowe et al., 2015) in sections 4 and 5 respectively. In section 6, we compare the results of joint deblending on simulated images of the Euclid, Rubin, and LSST surveys. We conclude in section 7.

2. Resampling and convolution of astronomical imaging

To maintain the flexibility of a pixelated galaxy model, combining surveys at different resolutions requires an explicit operation for resampling and convolution with the PSF. Both operations can be combined analytically.

2.1. Image modeling

Let the function $f : \mathbb{R}^2 \to \mathbb{R}$ represent the intrinsic, continuous surface brightness of the sky. Acquisition of images by telescopes causes the function $f$ to be convolved by the PSF $p$ of the telescope. The telescope is equipped with a camera with $n_{pix} \times n_{pix}$ pixels, which then integrates $(p \ast f)$ over the pixel area and samples it at the pixel positions $(x_k, y_k) \forall k \in [0, n_{pix}]^2$.

Ignoring all notions of noise in the acquisition process for the purpose of the demonstration, an image can therefore be described as a set of discrete samples of a continuous function $I$ defined as:

$$I(x, y) = (\text{rect}^{2D}_h \ast (p \ast f))(x, y), \quad (x, y) \in \mathbb{R}^2,$$

where $h$ is the linear size of a square pixel, i.e. $h = x_k - x_{k-1} = y_k - y_{k-1}$. The function $\text{rect}^{2D}_h$ is the 2-dimensional rectangle function defined by:

$$\text{rect}^{2D}_h(x, y) = \begin{cases} 
1 & \text{if } |x|, |y| < h/2, \\
0 & \text{otherwise}.
\end{cases}$$

Convolution by $\text{rect}^{2D}_h$ followed by sampling at the pixels center, is identical to the integration of the convolved surface brightness $(p \ast f)$ over the pixels surface.

2.2. Whittaker-Shannon interpolation

The Whittaker-Shannon interpolation states that a continuous, band-limited function, $f : \mathbb{R} \to \mathbb{R}$ can be recovered at any point $x$ from an infinite number of sampled evaluations $f_k = f(x_k)$:

$$f(x) = \sum_{k \in \mathbb{Z}} f_k \text{sinc}\left( \frac{x - x_k}{h} \right),$$

where $h$ is the sampling interval of points $x_k$, i.e. $x_k - x_{k-1} = h$ and $\text{sinc}$ is the normalized sinc function expressed as:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$ .

Equation 2 requires that $f$ is band-limited, which means that its Fourier transform $\hat{f}(\nu)$ has non-zero values only in the region $|\nu| \leq \frac{1}{2h}$. The Whittaker-Shannon interpolation formula trivially extends to two-dimensional functions:

$$f(x, y) = \sum_{(k_1, k_2) \in \mathbb{Z}^2} f_{k_1, k_2} \text{sinc}\left( \frac{x - x_{k_1}}{h} \right) \text{sinc}\left( \frac{y - y_{k_2}}{h} \right).$$

Since band-limited signals cannot be compact in real space, the summations in the previous equations have to be carried over the entire $\mathbb{Z}^2$ domain, which is not computationally tractable. In practice, summations are carried over a limited spatial domain, which is an approximation of the Whittaker-Shannon interpolation. With our target application being the modeling of individual astronomical objects (stars and galaxies), the approximation errors are limited by the fact that the flux of these objects is concentrated in a small region around their center.

In the following equations we will abstract the summation domain such that it is implied that all summations are over $\mathbb{Z}$ or $\mathbb{Z}^2$, unless otherwise specified.
2.3. Interpolation and convolution

Because the operations of convolution and resampling are computationally intensive, we seek a strategy to perform both as a single operation. We first consider the one-dimensional case of resampling and convolution of a function \( f : \mathbb{R} \rightarrow \mathbb{R} \) sampled at \( n \) positions \( x_k \), spaced regularly by a length \( h \) such that \( x_k = kh \). Our goal is to express the relation between these samples and the \( n' \) samples of the function \( f \ast p \) evaluated at \( n' \) linearly spaced positions \( X_l \) separated by \( h' \), where we generally assume but not require \( h' \neq h \). The convolution kernel \( p \) is arbitrary, with \( n_p \) samples at positions \( x_{nl} = mh' \). Throughout this calculation, we assume that \( f \) and \( p \) are band-limited for a grid with spacing \( h \). Using equation 2, we can represent function \( f \ast p \) at any point \( x \in [0, nh] \) as a function of the regular samples \( f_k = f(x_k) \) and \( p_m = p(x_m) \):

\[
(f \ast p)(x) = \sum_{k} f_k \text{sinc}\left(\frac{x-x_k}{h}\right) \ast \sum_{m} p_m \text{sinc}\left(\frac{x-x_m}{h}\right).
\]

With the Fourier transform of the sinc expressed as:

\[
\text{sinc}\left(\frac{x-x_k}{h}\right)(\nu) = h \times \text{rect}_h(\nu) e^{-\imath \nu x_k},
\]

where \( h \times \text{rect}_h(\nu) e^{-\imath \nu x_k} \) is the Fourier equivalent of a real shift by \( x_k \), we can write the Fourier transform of equation 4:

\[
(f \ast p)(\nu) = h^2 \sum_{k} f_k \text{rect}_h(\nu) e^{-\imath \nu x_k} \ast \sum_{m} p_m \text{rect}_h(\nu) e^{-\imath \nu x_m} \Rightarrow h^2 \text{rect}_h(\nu) \sum_{k} f_k \sum_{m} p_m e^{-\imath \nu (x_k+x_m)}.
\]

Note here that the product of two identical top-hat functions with the same width gives the same top-hat function. As a result, when transforming back into direct space, a factor \( h \) remains, such that

\[
(f \ast p)(x) = h \sum_{k} f_k \sum_{m} p_m \text{sinc}\left(\frac{x-x_k-x_m}{h}\right).
\]

By noticing that equation 8 defines a two-dimensional discrete convolution product by a shifted sinc function, it can be rewritten as:

\[
(f \ast p)(x, y) = h^2 \sum_{k} f_{k_1,k_2} \sum_{m_1,m_2} p_{m_1,m_2} \times \\
\text{sinc}\left(\frac{x-x_{k_1}-x_{m_1}}{h}\right) \text{sinc}\left(\frac{y-y_{k_2}-y_{m_2}}{h}\right).
\]

Equation 9 defines a convolution product with three terms, \( f \), \( p \) and \( S_h \), and it is linear in all three of them.

2.4. Resampling

We now combine the results of sections 2.1 and 2.3 and apply the convolution scheme to images with different resolutions. Let us consider two different images \( (I_1, I_2) \) of the same patch of the sky with two different resolutions, PSFs, and number of pixels, but in the same wavelength range. Following equation 1, these images are described by:

\[
I_1(x_{k_1}, y_{k_2}) = (\text{rect}_{h_1} \ast f \ast p_1)(x_{k_1}, y_{k_2})
\]

\[
I_2(x_{l_1}, y_{l_2}) = (\text{rect}_{h_2} \ast f \ast p_2)(x_{l_1}, y_{l_2}),
\]

where \( (k_1, k_2) \in [0, n_1] \times [0, n_2] \) and \( (l_1, l_2) \in [0, n_1'] \times [0, n_2'] \). We use the following convention, with the hope that this will save the readers a few headaches: When indexing samples on two different grids, upper-case letters will refer to samples on the coarse grid \( (X_l, Y_l) \) and lower case letters will refer to samples on the finer grid \( (x_k, y_k) \). That means \( h_1 \leq h_2 \).

The continuous functions \( p_1 \) and \( p_2 \) that describe each instrument’s PSFs are generally not known, and are instead represented by their integrated and sampled values:

\[
P_1(x_{ml}, y_{ml}) = (\text{rect}_{h_1} \ast p_1)(x_{ml}, y_{ml})
\]

\[
P_2(x_{ml}, y_{ml}) = (\text{rect}_{h_2} \ast p_2)(x_{ml}, y_{ml}).
\]

We also assume that the Fourier support of \( p_1 \) is larger than that of \( p_2 \), which is usually satisfied when \( h_1 \leq h_2 \) to ensure critical sampling of the PSFs.

While the images \( I_1 \) and \( I_2 \) could describe multiple observations, we are interested in fitting models that can have higher resolution and sharper PSF than the observations. For non-parametric models, the model itself can be represented as an image. For the sake of this demonstration, where images are assumed noiseless, we choose to parameterize our fiducial model of the sky as the higher-resolution image \( I_1 \), which means that on top of the inescapable rect convolution, we allow it to maintain its own effective PSF \( p_1 \). This has the added benefit to maintain the band-limitedness of the model required for our interpolation scheme. Due to the linearity of the convolution operation, and the ability to resample at any location, the combined operation can be expressed as a linear equation:

\[
I_2 = R_2 I_1
\]

With this in mind we can use equation 8 to build the operator \( R_2 \) in equation 13. Because the model, \( I_1 \), is a sampling of the integrated surface brightness of the sky in pixels of size \( h_1 \), convolved by PSF \( p_1 \), operator \( R_2 \) has to account for three effects:
the difference in PSF between \( p_1 \) and \( p_2 \),

- the resampling from positions \((x_k, y_k)\) to \((X_1, Y_1)\),

- and the difference in integration surface between \( I_1 \) and \( I_2 \).

To address the first item, we need to compute the difference convolution kernel \( P_d \) between \( P_1 \) and \( P_2 \). Using equation 12, we can express \( P_d \) as a function of \( p_1 \) and \( p_2 \):

\[
P_d = \mathcal{F}^{-1} \left( \frac{\text{rect}^{2D}_{h_2} * p_2}{\text{rect}^{2D}_{h_1} * p_1} \right),
\]

(14)

where \( \mathcal{F}^{-1} \) stands for the inverse Fourier transform. This ratio is well-defined if the Fourier support of \( p_1 \) is larger than that of \( p_2 \), which we already required earlier. It covers the particular case of the vanishing PSF \( p_1(x_m, y_m) = \delta(x_m, y_m) \) and that of the identity mapping where \( \text{rect}^{2D}_{h_2} * p_1 = \text{rect}^{2D}_{h_2} * p_2 \) leads to \( P_d = \delta \). To compute the difference kernel in practice, we need to have \( P_1 \) and \( P_2 \) sampled on the same grid. Because the grid of \( I_1 \) is chosen to be the reference grid and to prevent down-sampling of \( P_1 \), we build \( P_{d1} \), the interpolation of \( P_2 \) on \( P_1 \)'s grid:

\[
P^{(i)}_{d1}(x_{m1}, y_{m2}) = \sum_{o_{1,2}} P_{2(o_1,o_2)} S_{h_2}((x_{m1}, y_{m2}) - (X_1, Y_2)),
\]

(15)

\[
P^{(i)}_{d1} \text{ is the interpolation of } P_2 \text{ integrated in windows of size } h_2 \text{ but sampled in with pixel spacing } h_1.
\]

Using equation 9, we can interpolate the samples for \( I_1(x_{k1}, y_{k2}) \) at the locations of \( I_2 \)'s pixels by substituting \( p \) for \( P_d \) and \((x, y)\) for the set of \((X_1, Y_1)\) values:

\[
I_2(X_1, Y_1) = h_2^2 \sum_{k_{1,2}} \sum_{m_{1,2}} I_1(x_{k1}, y_{k2}) \text{sinc} \left( \frac{X_1 - x_{k1} - x_m}{h_1} \right) \text{sinc} \left( \frac{Y_1 - y_{k2} - y_m}{h_1} \right).
\]

(16)

It follows that operator \( R_2 \) can be represented by a \( n_1 \times n_2 \) matrix, whose elements are defined by the inner sum of the first line of equation 18, and quantifies the contribution of model sample \( I_{o_{1,2}} \) to image sample \( I_2(X_1, Y_1) \). This expression holds because the difference kernel carries the ratio of pixel integrations \( \text{rect}^{2D}_{h_2} / \text{rect}^{2D}_{h_1} \) between images of \( p_1 \) and \( p_2 \).

3. Accelerated resampling

According to equations 8 and 18, resampling and convolving an image requires one to compute the matrix \( R_2 \) of size \( (n_2^2 \times n_2^2) \) and to perform a matrix multiplication by \( R_2 \) every time the resampling and convolution is applied. We remind the reader that the target application for this scheme is iterative model fitting with \( \text{SCARLET} \), which implements a gradient descent optimisation. The goal of the acceleration presented here is therefore to minimise the time spent on computations at each iteration. We seek to reformulate equation 8 to accelerate these operations, which should be possible due to the separability of the sinc function.

3.1. The case of non-rotated grids

Let a high resolution image \( I_1 \) be of size \( n_1 \times n_1 \) and its low resolution counter-part \( I_2 \) with size \( n_2 \times n_2 \), verifying equation 18. First, we notice that there are in total \( n_1 \) different values of pixel coordinates \((x_k, y_k)\) for \( I_1 \) on its x-axis and \( n_1 \) different values on its y-axis. Similarly, in the case where the grids of \( I_2 \) and \( I_1 \) are aligned, there are only \( n_2 \times n_2 \) different values for the pixel coordinates \((X_k, Y_k)\) of \( I_2 \) along its x and y-axes respectively. As a result, it is only necessary to compute \( n_2 \times n_2 \times n_1 \) evaluations of 1-dimensional sinc kernels to determine the \( n_2 \times n_2 \times n_1 \times n_1 \) matrix \( R \). This realization allows a faster computation of \( R \), but it is possible to achieve even faster resampling by separating the sinc kernel in its 1-dimensional components and factoring it such that equation 18 applied to \( I_2 \) and \( I_1 \) becomes:

\[
I_2(X_1, Y_1) = h_2^2 \sum_{k_{1,2}} \sum_{m_{1,2}} I_1(x_{k1}, y_{k2}) \text{sinc} \left( \frac{X_1 - x_{k1} - x_m}{h_1} \right) \text{sinc} \left( \frac{Y_1 - y_{k2} - y_m}{h_1} \right).
\]

(18)

The second line defines \( n_2 \), one-dimensional convolution products by sinc functions shifted by \( Y_1 \). The first line also contains one-dimensional convolution products along the x-axis of image \( I_1 \), by sinc kernels shifted by \( X_1 \). The first two summations define a matrix product between the results of the two one dimensional convolutions. A more compact expression for equation 18 is therefore:

\[
I_2(X_1, Y_1) = h_1^2 \sum_{y_2} \sum_{x_1} (I_1 \circ s_1)(X_1 - x_1, y_2)(P_{d1} \circ s_1)(x_1, Y_1 - y_2),
\]

(19)

(\( \circ \) stands for the one-dimensional convolution \( \ast \) and \( s_1 \) is the 1-dimensional sinc function defined as \( s(x) = \text{sinc}(\frac{x}{h_1}) \). Equation 19 defines two convolution products. However, given that samples \((X_k, Y_k)\) and \((x_k, y_k)\) are not on the same grid, fast algorithms for convolutions do not apply.

By evaluating these convolutions and shifts in the Fourier domain, using the rectangle function as the analytic Fourier transform of the sinc, we find that this scheme allows us to perform resampling ten times faster than when using the full resampling matrix \( R \) on typical runs with a high resolution image of size \((250 \times 250)\) and its low resolution counterpart with size \((50 \times 50)\). Also, the setup time spent in computing matrix \( R \) on typical runs is divided by a factor \( \approx 100 \) by only computing one-dimensional convolutions and shifts on \( P_d \).

Here, we chose to associate the sinc with arguments along the x-axis with function \( I_1 \) and the sinc with arguments along the y-axis to \( P_d \) in equation 18, however we could just as easily have inverted this choice. In cases where images have rectangular shapes (meaning \( n_1 \neq n_1 \)), this freedom allows us to optimize the number of operations by choosing to convolve \( I_1 \) by a sinc along its smallest dimension.
3.2. Generalization to rotated frames

Two images of the same patch of sky might happen to be aligned, but in general this assumption is not true. The technique we just described is a special case of a more general formalism that we introduce in this section. In the case where images are rotated with respect to one another, the number of different values $X_i$ and $Y_i$ can take is not limited to $n_{2x}$ and $n_{2y}$, but can be as large as $n_{2x} \times n_{2y}$ for each of them. This is illustrated in Figure 1, where the rotated images on the right-hand-side show pixel positions $X_i$ and $Y_i$ that have nine different values each, while aligned images as represented on the left have three different values each.

In the rotated case, using the splitting outlined in section 3.1 leads to large matrix multiplications and is actually detrimental to the speed of the operation, making it a worse formulation than the naive equation 8.

We can find a more general formulation for the factorization of equation 8 by:

- firstly, transforming coordinates $(X_i, Y_i)$ to an un-rotated frame where all pixel coordinates $X_i$ and $Y_i$ can be expressed by $n_{2x}$ and $n_{2y}$ different values respectively.
- secondly, writing equation 8 as the three-factor convolution product that it is, we can use the associative property to split and redistribute the arguments of the discrete convolution along carefully selected axes.

3.2.1. Frame rotation

Because $I_2$ is acquired on a regular grid, there should exist a frame in which its coordinates take a minimal number of different values $(X_m, Y_n)$. This frame is trivially given by applying to the coordinates $(X_i, Y_i)$ of $I_2$, the rotation matrix which angle matches the angle $\theta$ between both images such that:

$$X_i = X_m \cos \theta + Y_n \sin \theta,$$
$$Y_i = Y_n \cos \theta - X_m \sin \theta.$$

3.2.2. Convolution associativity

By realizing that equation 8 is a three-factor discrete convolution that generates a result sampled on a different grid than two of its factors, we can write $I_2$ as:

$$I_2 = I_1 \ast P_d \ast S_h$$

By associativity and commutativity of the convolution product and by substituting equation 20, we can rewrite equation 8 as:

$$I_2(X_i, Y_i) = h^2 \sum_{k_1, k_2} I_1(k_1, k_2) \sum_{m_1, m_2} P_{d,m_1,m_2} S_{m,n}$$
$$= h^2 \sum_{k_1, k_2, m_1, m_2} P_d(x_{k_1}, y_{k_2}) \sin\left(\frac{y_{k_2} + y_{m_2}}{h}\right) \times$$
$$\sum_{m_1} P_{m,d}(x_{m_1}, y_{m_2}) \sin\left(\frac{x_{k_1} + x_{m_1}}{h}\right).$$

with:

$$S_{m,n} = \sin\left(\frac{X_m \cos \theta + Y_n \sin \theta - x_{k_1} - x_{m_1}}{h}\right)$$
$$\sin\left(\frac{Y_n \cos \theta - X_m \sin \theta - y_{k_2} - y_{m_2}}{h}\right)$$

$$F_d(x_{k_1}, y_{k_2}) = I_1(x_{k_1} - \frac{Y_n}{h} \sin \theta, y_{k_2} - \frac{Y_n}{h} \cos \theta)$$
$$P_d(x_{m_1}, y_{m_2}) = P_d(x_{m_1} - \frac{X_m}{h} \cos \theta, y_{m_2} + \frac{X_m}{h} \sin \theta).$$

In the last equality of equation 22, the first term in $I_1$ is computed by shifting image $I_1$ in the Fourier domain in $n_{2x}$ different directions $(Y_n \sin \theta, Y_n \cos \theta)$. The remainder is obtained by performing $n_{2x}$ shifts of the result of the convolution between $P_d$ and $S_h$ by vectors with components $(X_m \cos \theta, -X_m \sin \theta)$.

Once again, we chose to pick the terms in $Y_n$ for the shifts in $I_1$, but an equivalent result would be achieved by having $I_1$ shifted by $(X_m \cos \theta, -X_m \sin \theta)$ and $P_d \ast S_h$ shifted by $(Y_n \sin \theta, Y_n \cos \theta)$. While it might seem unsettling to see terms in $Y_n \sin \theta$ associated with the $x$-axis of $I_1$, the reader should keep in mind that this term comes from the decomposition of $x_{m_1}$. It is also important that both axes be shifted by values depending on the same “unrotated” variable, $(Y_n$ for the shift in $I_1$, in equation 22), otherwise there exists as many combinations of $(X_m, Y_n)$ as there are pixels in $I_2$ and the computational gain is lost.

3.2.3. Link with the aligned case and extensions

The acceleration from section 3.1 qualifies easily as a particular case of the scheme described in section 3.2. The main semantic difference between these formulations is that, in the former we used the separability of the sinc function to perform the dimensionality splitting, while in the latter, we reformulated the discrete convolution product. Actually, the result from equation 19 can be obtained by using the same convolution reformulation and then factoring by the one-dimensional sinc factors. Alternatively, in equation 22, it is possible to factorize one of the one dimensional sinc factors and have $I_1$ convolved by this sinc.

By setting the value of $\theta$ to zero, equations 19 and 22 are equivalent, if not for the choice of the axis along which the shift is operated. We choose here to show both approaches of sinc factorization and convolution reformulation for two reasons. First, we think that the factorization formulation makes an easier introduction to the notion of dimension splitting for acceleration and would thus ease the understanding of subsection 3.2. Second, because this separation reflects the content of our implementation in scarlet. While both formulations are strictly equivalent and lead to the same result when $\theta = 0$, the convolution reformulation in its implementation suggests 2-dimensional Fourier transforms with 2-dimensional shifts, one of them being of magnitude 0. This requires twice as many operations than with the factorization formulation where the Fourier transform is applied along one dimension only and the number of multiplications for the shift is doubled.

The dimensional splitting we deployed in this section to perform fast convolutions onto rotated frames can actually be generalized to any stationary, separable transform on the grid.
Indeed, the matrix rotation used in equation 20 can easily be replaced by a matrix with arbitrary coefficients as long as they are independent on the grid’s coordinates, or at least as long as they don’t introduce cross terms that depend on both $X_m$ and $Y_n$. This extends to polynomial transformations with the condition that they contain no rectangular terms in $X_m^a \times Y_n^b$ and can be applied to conventional convolutions, including interpolations, and does not require the separability of the kernel.

4. Speed and complexity

To test the performance of our algorithm, we implemented it in the scarlet framework, and compare the run-time and the interpolation accuracy with a thoroughly tested reference implementation for resampling: galsim.

Implementing the formulation described above requires several operations, whose computational complexity depends on the number of pixels in the high- and low-resolution frames. In this section we compute the complexity of these operations for square images with $M \times M$ pixels in the low-resolution channel and $N \times N$ in the high-resolution channel. Given that this algorithm is meant for an iterative framework, we make a difference between setup operations that are performed once, and running operations that have to be performed every time the model changes.

Setup operations comprise:

- Fourier transform of the low- and high-resolutions PSFs: $O(M^2 \log(M)) + O(N^2 \log(N))$
- $N$ shifts of the low-resolution PSF in the Fourier domain (multiplication by a complex exponential along one direction: $O(NM^2)$
- inverse Fourier transforms of the shifted PSFs: $O(NM^2 \log(M))$,
- matrix multiplications by the sinc kernel along each axis: $O(M^2N^2)$.
Run operations execute these steps every iteration:

- Fourier transform the high-resolution image along one axis: $O(N^2 \log(N))$,
- apply $M$ shifts in the Fourier domain: $O(MN^2)$
- apply inverse FFT to each shifted image along one axis: $O(MN^2 \log(N))$
- multiply by $P_d$ to obtain the low-resolution image $O(M^2 N^2)$

The total cost of which is therefore

$$C_s = O(N^2 (M + 1)(\log(N) + M)) \quad (23)$$

The complexity is dominated by the term in $M^2 N^2$ which is expected for the Whittaker-Shannon interpolation. While this scheme does not reduce the complexity, we implement it as a matrix multiplication between two matrices with a total number of elements $MN^2$ each. A classical interpolation, where the kernel is applied without separation, would amount to multiplying a matrix with $M^2 N^2$ elements by a $N^2$ elements vector. While these two strategies require the same number of operations, our approach has a smaller memory footprint as we do not need to store a large $M^2 N^2$ elements matrix. In practice, this leads to a 10-fold speed up on a personal laptop.

To assess the performance of our resampling scheme, we compare it against the resampling scheme described in BG and implemented in the galsim package as the default scheme. In their approach, BG use an approximation of the Whittaker-Shannon interpolation by a quintic kernel with only 6 elements along each dimension, applied in Fourier space, and rely on 4-fold padding of the input image to limit the interpolation errors. Convolution is then performed with galsim’s default setting in a separate operation at the resolution of the resampled image in the Fourier domain. For comparison, in the resampling scheme from BG, the authors report a complexity of:

$$C_g = O(N^2(k^2 + s^2 \log(sN)) + M^2(3 + \log(M))) \quad (24)$$

where $k = 6$ is the size of the preferred quintic interpolation kernel and $s = 4$ is the default 4-fold padding necessary to prevent the formation of ghosts in the image. Comparing equations 23 and 24 suggests a theoretical factor of a few hundreds between the run-times of our implementation and that of galsim’s default for reasonable values of $M$ and $N$.

We compare the actual run-time, without setup, of our implementation, $t_s$, with the run-time for the BG scheme in galsim’s default, $t_g$. To do so, we generate images at 5 different resolutions to emulate observations with current or upcoming telescopes. The PSFs used in these simulations are Gaussian profiles. Pixel scales and PSF sizes (standard deviations) are given in Table 1.

| Telescope  | $\sigma_{\text{psf}}$ (arcsec) |
|------------|-----------------|---|---|
| HST        | 0.06            | 0.016 |
| EUCLID     | 0.1             | 0.186 |
| ROMAN      | 0.11            | 0.186 |
| HSC        | 0.167           | 0.62 |
| RUBIN      | 0.2             | 0.7  |

Here, each observation that has a larger pixel size than another, also has a larger PSF, meaning that comparing resolutions is consistent. For each telescope, except Rubin, we generate seven high-resolution images with 20, 30, 50, 70, 100, 150 and 200 pixels on a side. For a given observation, we then resample the image on a grid that spans the same area on the plane of the sky but at the resolution of all telescopes that have a larger pixel size using either our new or the galsim resampling schemes. The results of this comparison are shown in Figure 2, where we see that the dependence between both strategies is in fact well described by a linear model:

$$t_s/t_g \approx -0.17 + 0.048M. \quad (25)$$

The relative complexity does appear dominated by a term $\propto M$ instead of $M^2$ as suggested by equations 23 and 24. This is most likely explained by internals of the galsim package that we have not accounted for, but we cannot provide a compelling explanation at this stage. While our algorithm is still prohibitively slow for large images compared to galsim, it is reassuring that its performance in a python implementation is within a factor of a few of the current state-of-the-art algorithm.

For reference we also show in Figure 3 the average timings of scarlet and galsim’s sinc and quintic resamplings.
We observe that our implementation is faster than galsim’s sinc implementation for scenes where the high resolution image is smaller than 200 pixels. For larger images timing seems comparable. For future works, it might be of interest to implement approximate interpolation schemes to reduce computation times. For separable kernels, the acceleration presented in Section 3 would still provide benefits, thus enabling larger kernels that would mitigate approximation errors to remain computationally tractable.

All the figures presented in this section can be reproduced from notebooks Timing_tests and Timing_fit.

5. Resampling accuracy

To test the quality of the reconstructions enabled by our implementation, we compare it with reconstructions from galsim on simulated images. We use galsim to simulate a set of 10,000 images of real galaxy profiles from the COSMOS sample (Mandelbaum et al., 2012, 2014), at the resolution of the emulated surveys from Table 1. Each of these images is drawn on a 60 × 60 pixels grid, chosen to represent extended complex galaxy profiles that can be resampled in a timely fashion. These images serve as low resolution images, which are the ground truth for our reconstructions. For each of them, we use galsim to produce images of the galaxies with the PSF and samplings of higher resolution surveys.

We then use our method and galsim’s default quintic interpolation to resample the high-resolution images to their low-resolution version and compare the result with the ground truth. While both ground truth images and galsim resampled images are produced using the same software, the way they are produced is fundamentally different. Ground truth images are drawn directly from galsim’s GSObjects. Resampled images are interpolated from the pixels of the high resolution images. For reference, we also reconstruct low resolution images using galsim’s implementation of the sinc interpolation.

For each reconstructed low resolution image, we compute the source distortion ratio of the reconstruction defined as:

$$ SDR(\hat{X}) = 10 \log_{10} \left( \frac{||X||}{||\hat{X} - X||} \right) $$  \hspace{1cm} (26)

where $\hat{X}$ is the reconstructed low resolution image, $X$ is the true low resolution image and $||.||$ is the Euclidean 2-norm. The higher the SDR, the more accurate a reconstruction is.

We show the SDRs for the scarlet and the galsim reconstructions averaged over the different galaxy realizations in Figures 4 and 5. The average SDR difference is always positive, indicating that the scarlet’s resampling accuracy exceeds galsim’s default resampling in every pair of surveys. The most obvious differences are seen in the diagonal elements. They correspond to a resampling of a survey image on itself. In this case, scarlet reproduces the identity operator but galsim does not. As the ratio between pixel sizes increases, the galsim’s default scheme tends to yield worse SDRs on average. SDRs for scarlet seem rather uniformly high without any noticeable dependence between surveys. This is to be expected as the reconstruction method in galsim is an approximation, while scarlet’s reconstructions use a perfect interpolation scheme in the limit of band-limited signals. Results from galsim’s sinc resamplings are shown in figure 6 and show similar quality of reconstruction. Only along the diagonal axis that represents the identity mapping do we notice significantly different results, but even there, galsim’s sinc interpolation shows far better SDR than in the default case. Our findings are confirmed by a visual inspection of the resampling residuals in Figure 7.

We notice that the variance of scarlet’s SDR measurements is high and can be larger than the difference between the average for scarlet and galsim measurements. After further visual inspection, we realised that in some cases, the SDR for scarlet reconstructions can be smaller than those of galsim measurement. An example is shown in Figure 8 that reveals evident edge truncation, where flux of the galaxy extends to and possibly beyond the edge of the image. It affects both reconstructions, but the scarlet scheme more so than galsim. Spatially finite extent is formally the only approximation we make in our scheme because it is practically impossible to extend the summation in equation 3 over $Z^2$. The problem arises in less than ~ 10% of the images in the HSC to Rubin, Roman to HSC and Roman to Rubin cases. For the rest of the survey pairs, galsim is better than scarlet in less than 1% of the cases. We note that the size of the images was fixed at 60 × 60 pixels for this test, which at least in some cases appears too restrictive to achieve higher SDR, and can be mitigated by additional padding at the expense of increased runtime.

Figures 4 and 5 can be reproduced using the notebook Reconstruction_comparison. Figures 7 and 8 can be reproduced using the GSObject_comparison notebook.

---

3https://github.com/herjy/Multi_resolution_comparison/blob/master/Timing_test/Time%20fit.ipynb
4https://github.com/herjy/Multi_resolution_comparison/blob/master/Timing_test/Time%20fit.ipynb
5https://github.com/herjy/Multi_resolution_comparison/blob/master/Reconstruction_test/Reconstruction_comparison.ipynb
6https://github.com/herjy/Multi_resolution_comparison/
Figure 4: Mean SDRs of the scarlet (left) and galsim (right) reconstructions for each couple of resolutions. The y-axis shows the high resolution observation from which the reconstruction is built. The x-axis shows the target resolution. Initials HR and LR stand for high resolution and low resolution respectively.

Figure 5: Standard deviation of the SDRs of the scarlet (left) and galsim’s default (middle) reconstructions. Right hand-side panel shows the difference between the mean scarlet and galsim SDRs as shown in Figure 4.

Figure 6: Mean and standard deviation of the SDR from galsim’s sinc interpolation (right and middle panels respectively). Right hand-side panel shows the difference between the mean scarlet and galsim sinc interpolation SDRs.

6. Joint deblending

A primary application for our resampling method is the joint modeling and deblending of galaxies by simultaneously fitting
the imaging data of multiple surveys. We refer to Melchior et al. (in prep.) for a review of astronomical blending and Chary et al. (2020) for a recent overview of the scientific benefits and operational considerations of joint survey processing.

Specifically, we model multiple galaxies in simulated Euclid and Rubin images with scarlet. In brief, scarlet assumes that a given patch of the sky contains a predefined number $K$ of sources, each of which is described by a broad-band SED, $A_k$, and a non-parametric model of the morphology, $S_k$. This yields a model

$$M = \sum_k A_k^T \times S_k = A \cdot S$$

of the data in every band. For multi-instrument modeling, the vector $A_k$ contains the concatenation of the amplitudes of source $k$ in all bands of each available survey. It is model $M$, whose single-band images $M_b$ will act as the high-resolution image $I_1$ in equation 13, which is resampled and convolved to the resolution and PSF of the respective surveys. The model parameters $A$ and $S$ are then iteratively optimized by adaptive constrained gradient descent (Melchior et al., 2019). Further details on scarlet can be found in Melchior et al. (2018).

6.1. Simulations setup

The simulations were produced with the galsim package. Each Rubin stamp has 60 × 60 pixels, which corresponds to 118 × 118 pixels for the Euclid images. We inserted between 1 and 10 galaxies in each stamp. The positions of galaxies within the stamp are uniformly distributed, their colours are drawn from the catalog simulation framework Catsim\textsuperscript{7}, also used in the construction of the Cosmos DC2 simulations (Korytov et al., 2019). This catalog was generated for the BlendingToolKit\textsuperscript{8} package and can be downloaded at this address\textsuperscript{9}. We artificially draw random magnitudes from a uniform distribution between 20 and 29 for every galaxy. While unrealistic, this step ensures that we can explore the full range of relative blending impacts with a similar number of sources.

Galaxy morphologies are drawn from galsim’s COSMOS catalog of real galaxy images. These galaxy images are convolved with a gaussian kernel with a 2-pixel-wide full width at half maximum. This operation ensures that the ground truth galaxies are band-limited. Galaxy positions are drawn at random across the stamps. Individual galaxy images in a given stamp are summed up together before adding noise. We use Gaussian profiles as PSFs with the appropriate width for both surveys and band wavelength range. The details of the parameters used in the simulations are in table 2.

Each individual galaxy image is saved for comparison with the models reconstructed by scarlet. For consistency in the comparison, each image was normalised to maximum pixel value of 1 when saved.

In the resulting simulations, we sacrificed the realism of the scenes to instead homogeneously probe a large number of configurations in terms of magnitudes across bands, galaxy numbers and morphologies.

---

\textsuperscript{7}https://www.lsst.org/scientists/simulations/catsim
\textsuperscript{8}https://github.com/LSSTDESC/BlendingToolKit
\textsuperscript{9}https://stanford.app.box.com/s/1nzjlinejqandduykyjeyxtyylbk
| bands | u | g | r | i | z | y | Euclid VIS |
|-------|---|---|---|---|---|---|------------|
| Pixel scale (arcsec) | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.010 |
| PSF width (arcsec) | 0.327 | 0.31 | 0.297 | 0.285 | 0.276 | 0.267 | 0.16 |
| exposure time (s) | 1680 | 2400 | 5520 | 5520 | 4800 | 4800 | 2260 |
| zero point | 9.16 | 50.70 | 43.70 | 32.36 | 22.68 | 10.58 | 6.85 |
| sky level | 22.9 | 22.3 | 21.3 | 20.5 | 19.6 | 18.6 | 22.9 |

Table 2: Parameters used to generate simulations of Euclid and Rubin blended scenes.

6.2. Deblending accuracy

Deblending is a complex procedure that depends on accurate detection and, in the case of iterative non-linear solvers, initialization of every source. In this paper, we are only interested in resampling and the new multi-instrument modelling capabilities it enables. Therefore, we use the ground truth for all source positions and initialise each galaxy morphology model with the Sérsic fit to each galaxy image from galsim. The spectra of each galaxy is initialized through scarlet, by solving the linear inverse problem of estimating spectra knowing the initial morphology. We will discuss realistic detection and initialisation schemes for joint-survey analyses in a forthcoming paper. We generated 11625 galaxy profiles in 2315 patches as described above.

We model each galaxy’s morphology with scarlet’s ExtendedSource class, i.e. with an image whose pixels satisfy positivity and monotonicity constraints as described in (Melchior et al., 2018). The models maintain a minimal PSF $p_1$ with a Gaussian profile of standard deviation 0.8 pixels in the respective frame.

For the comparison, the true galaxy profiles are rendered with galsim on the Euclid and Rubin grids with their respective pixel integrations and convolved with $p_1$. We then compute the SDR between the scarlet model and the true image from galsim. Each stamp is modeled in three different ways:

- **low-resolution only**: All six Rubin images are modeled at the Rubin resolution.
- **high-resolution only**: The Euclid VIS image is modeled at the Euclid VIS resolution.
- **joint modeling**: Both Rubin and Euclid images are modeled simultaneously at the Euclid resolution, using our resampling scheme to match the Rubin data.

In figure 9, we show the SDR of the reconstructed images as a function of the galaxies’ magnitude in each pair of Rubin-Euclid images. Given that Rubin data have multiple bands, the magnitude used as a reference in these figures is the smallest magnitude for each galaxy across all bands. This lets us probe the maximum impact of using brighter images in the joint resolution modeling. By probing uniformly the range of magnitude from 20 to 30 across all bands and resolutions, we voluntarily create a fraction of images with unrealistic photometry. Given the spectral overlap between Euclid and Rubin’s bands, it is unlikely that a galaxy could be very faint in Rubin images and very bright in Euclid (bottom right corners of figure 9). We choose to leave these unrealistic cases in our study because they let us probe the advantage of combining different surveys with different wavelength ranges, where larger magnitude differences could occur. It is evident that each survey individually produces models that are most accurate whenever the sources are bright in that survey’s bands. However, joint modeling yields high accuracy when the sources are bright in any of the surveys.

For better interpretation, we present the difference between the SDRs of the single-resolution cases and the joint modeling in figure 10. The fact that virtually all magnitude bins are colored in red shows that joint modeling yields globally superior
reconstruction accuracy. Unsurprisingly, the biggest improvements, of up to an order of magnitude, are from cases where one of the surveys has images that are substantially brighter than the other; e.g. a bright blue source that is easily observed by Rubin, but not by Euclid’s redder VIS filter. We want to point out that, while unrealistic in its design, this test provides for the first time clear evidence of the improvement in galaxy modeling from joint pixel-level analysis of blended galaxies.

All the figures presented in this section can be reproduced from the notebooks found at [GitHub](https://github.com/berjy/Multi-resolution_comparison/tree/master/Reconstruction_Jest).

7. Conclusion

We introduce a novel method to express convolution and resampling of one- and two-dimensional functions as a single operation directly based on the Whittaker-Shannon interpolation formula. Applied to astronomical images it allows one to render a high-resolution image with a given PSF to a lower-resolution image with a different PSF. The assumptions for this method are that the high-resolution image is critically sampled, the amount of light at the edge of the high-resolution image is negligible, and the PSF of the high-resolution image is narrower than that of the low-resolution image.

We compare our resampling scheme with the method by Bernstein & Gruen (2014), implemented in galsim, which uses a much more compact approximated interpolation kernel. We find that, with the accelerated version of our scheme, the runtimes of the resampling and convolution operation are comparable for very small images of $50 \times 50$ pixels and, for images of $200 \times 200$ pixels, a factor of $\sim 10$ slower. The most costly operation is a matrix multiplication, for which GPU architectures should provide substantial performance gains. However, we consistently find more accurate interpolation results, with residuals typically being reduced by an order of magnitude compared to galsim.

We implement this method in the galaxy modeling framework scarlet where it enables us to simultaneously fit images from different instruments. To our knowledge, this work presents the first application of non-parametric multi-band modeling across multiple instruments. We apply this new capability to the pressing problem of deblending deep astronomical imaging surveys, specifically those planned with the Euclid mission and the Vera C. Rubin Observatory. We find that joint modeling increases the reconstruction accuracy, as quantified by the source distortion ratio SDR of the morphological model, by up to an order of magnitude compared to analyzing both survey images separately. This study is, to our knowledge, the first investigation of the gains of joint pixel-level analysis for galaxies with a wide range of blending configurations. Follow-up studies can use our codes to investigate the accuracy of joint reconstructions for other quantities of interest, such as shape, photometry, and colour measurements. Simulation efforts to produce highly realistic images for upcoming surveys (Troxel et al., 2019; Korytov et al., 2019, e.g.) will be instrumental for reaching this goal.

In forthcoming works, we will continue the outlined path towards practically feasible pixel-level analysis of multiple surveys by addressing aspects we ignored in this work, in particular the problem of source detection and association across multiple instruments, the impact of astrometric and photometric calibration errors, and the initialisation and modeling of complex morphological patterns.

8. Acknowledgements

This paper makes use of software developed for the Large Synoptic Survey Telescope. We thank the LSST Project for making their code available as free software at [http://dm.lsst.org](http://dm.lsst.org).

This material is based upon work supported by the National Science Foundation under Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Con-
The results presented in this paper are reproducible using three packages developed by the authors. The versions of these packages as used in this publication are the following: matplotlib (Hunter, 2007), autograd and proxmin packages. The figures presented in this paper were produced using the matplotlib library (Hunter, 2007). The results presented in this paper are reproducible using three packages developed by the authors. The versions of these packages as used in this publication are the following: matplotlib (Hunter, 2007), autograd and proxmin packages. The figures presented in this paper were produced using the matplotlib library (Hunter, 2007).
of Multi-probe data from WFIRST, Euclid, and LSST. **MNRAS, 51, 201.** Rhodes, J., Nichol, R. C., Aubourg, É., Bean, R., Bouriginy, D., Bremer, M. N., Capak, P., Cardone, V., Carry, B., Conselice, C. J., Connolly, A. J., Cuillandre, J.-C., Hatch, N. A., Helou, G., Hemmati, S., Hildebrandt, H., Hložek, R., Jones, L., Kahn, S., Kiessling, A., Kitching, T., Lupton, R., Mandelbaum, R., Markovic, K., Marshall, P., Massey, R., Maughan, B. J., Melchior, P., Melli, Y., Newman, J. A., Robertson, B., Sauvage, M., Schrabback, T., Smith, G. P., Strauss, M. A., Taylor, A., & Von Der Linden, A. (2017). Scientific Synergy between LSST and Euclid. **ApJS, 233, 21. doi:10.3847/1538-4365/aa96b0. arXiv:1710.08489.**

Rowe, B., Hirata, C., & Rhodes, J. (2011). **OPTIMAL LINEAR IMAGINE COMBINATIONS.** *The Astrophysical Journal, 741, 46. URL: https://doi.org/10.1088%2F0004-637x%2F741%2F1%2F46. doi:10.1088/0004-637x/741/1/46.*

Rowe, B. T., Jarvis, M., Mandelbaum, R., Bernstein, G. M., Bosch, J., Simet, M., Meyers, J. E., Kacprzak, T., Nakajima, R., Zuntz, J., Miyatake, H., Dietersch, J. P., Armstrong, R., Melchior, P., & Gill, M. S. S. (2015). **GALSIM: The modular galaxy image simulation toolkit.** *Astronomy and Computing, 10, 121–150. doi:10.1016/j.ascom.2015.02.002, arXiv:1407.7676.**

Schuhmann, R. L., Heymans, C., & Zuntz, J. (2019). Galaxy shape measurement synergies between LSST and Euclid. *arXiv e-prints.* (p. arXiv:1901.08586. arXiv:1901.08586.)

Scolnic, D. M., Lochner, M., Gris, P., Regnault, N., Hložek, R., Aldering, G., Allam, S., C. T., Terek, Awan, H., Busas, R., Blazek, J., Chang, C., Gawiser, E., Goobar, A., Hook, I. M., Jha, S. W., Meiksin, J. D., Mandelbaum, R., Marshall, P., N. I., Nielsen, E., Rhodes, J., Rothchild, D., Sevilla Noarbe, I., Slosar, A., & Yoachim, P. (2018). Optimizing the LSST Observing Strategy for Dark Energy Science: DESC Recommendations for the Deep Drilling Fields and other Special Programs. *arXiv e-prints.* (p. arXiv:1812.00516. arXiv:1812.00516.)

Snodgrass, C., Carry, B., Berthier, J., Eggel, S., Momment, M., Petit, J. M., Spoto, F., Granvik, M., Laureijs, R., Altieri, B., Varek, R., Conversi, L., Nucaita, A., Popescu, M., Verdoes Kleijn, G., Kidger, M., Jones, G. H., Oszkiewicz, D., Juric, M., & Jones, L. (2018). Simultaneous LSST and Euclid observations - advantages for Solar System Objects. *arXiv e-prints.* (p. arXiv:1812.00607. arXiv:1812.00607.)

Spergel, D., Gehrels, N., Baltay, C., Bennett, D., Breckerinridge, J., Donahue, M., Dressler, A., Gaudi, B. S., Greene, T., Guyon, O., Hirata, C., Kalirai, J., Kasdin, N. J., Macintosh, B., Moos, W., Perlmutter, S., Postman, M., Rauscher, B., Rhodes, J., Wang, Y., Weinberg, D., Benford, D., Hudson, M., Jeong, W. S., Melchior, P., E., Traub, W., Yamada, T., Capak, P., Colbert, J., Masters, D., Penny, M., Savransky, D., Stern, D., Zimmermann, N., Barry, R., Bartusek, L., Carpenter, K., Cheng, E., Content, D., Dekens, F., Demers, R., Grady, K., Jackson, C., Kuan, G., Kruk, J., Melton, M., Nemati, B., Parvin, B., Poberzhzskiy, I., Peddie, C., Ruffa, J., Wallke, J. K., Whipple, A., Wollack, E., & Zhao, F. (2015). Wide-Field InfrarRed Survey Telescope Telescopes-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report. *arXiv e-prints.* (p. arXiv:1503.03757. arXiv:1503.03757.)

Street, R. A., Lund, M. B., Donachie, M., Khakpash, S., Golovich, N., Penny, M., Bennett, D., Dawson, W. A., Pepper, J., Rabus, M., Szkody, P., Clarkson, W. I., Di Stefano, R., Rattenbury, N., Huntephurmark, M. P. G., Tsapras, Y., Ridgway, S., Stassin, K., Bovza, V., Bhattacharya, A., Calchi Novati, S., & Shvartzvald, Y. (2018). Unique Science from a Coordinated LSST-WFIRST Survey of the Galactic Bulge. *arXiv e-prints.* (p. arXiv:1812.04445. arXiv:1812.04445.)

Truex, M., Long, H., Hirata, C., Choi, A., Jarvis, M., Mandelbaum, R., Wang, K., Yamamoto, M., Hemmati, S., & Capak, P. (2019). A synthetic wFIRST high-latitude imaging survey: Simulation suite and the impact of wavefront errors on weak gravitational lensing. *arXiv preprint arXiv:1912.09481.*

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering, 13, 22–30.*

Virman, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournopeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E. W., Vand ePlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., & Contributors, S. . . (2020). **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.** *Nature Methods, 17, 261–272. doi:https://doi.org/10.1038/s41592-019-0686-2.**