Astronomical Image Simulation for Telescope and Survey Development

Benjamin M. Dobke,[1] David E. Johnston,[2] Richard Massey,[3] F. William High,[4] Matt Ferry,[5]
Jason Rhodes,[1,5] and R. Ali Vanderveld[1,5]

ABSTRACT. We present the *simage* software suite for the simulation of artificial extragalactic images, based empirically around real observations of the Hubble Ultra Deep Field. The simulations reproduce galaxies with realistic and complex morphologies via the modeling of UDF galaxies as shapelets. Images can be created in the B, V, i and z bands for both space- and ground-based telescopes and instruments. The simulated images can be produced for any required field size, exposure time, PSF, telescope mirror size, pixel resolution, field star density, and a variety of detector noise sources. It has the capability to create images with either a predetermined number of galaxies, or one calibrated to the number counts of preexisting data sets such as the *HST* COSMOS survey. In addition, simple options are included to add a known weak gravitational lensing signal (both shear and flexion) to the simulated images. The software is available in IDL and can be freely downloaded for scientific, developmental, and teaching purposes.

1. INTRODUCTION

In the next decade, the quantity of data available to cosmology will rapidly increase. New telescopes, both on the ground and in space, promise to image many thousands of square degrees. The cosmology community is now tasked with developing methods to analyze such data, and to extract as much information from various astronomical phenomena as possible. These methods need to achieve unprecedented precision if the promise (and potential statistical power) of future surveys is to be fully exploited.

Developing image analysis tools requires realistic mock data, containing as many instrumental effects as possible, plus a known, underlying cosmological signal, against which measurements can be judged. To generate galaxy shapes in simulated ground-based images, *Skymaker* (Erben et al. 2001; Bertin 2009) uses a simple physical model of concentric isophotes with a de Vaucouleurs profile for elliptical galaxies, and an additional exponential component for spirals. By varying the model parameters, one can generate an unlimited number of unique simulated galaxies. However, deep field images from space-based telescopes contain galaxies with features more complex than these smooth analytical models can reproduce. Here we present the full *simage* pipeline (as gradually developed in Massey et al. 2004, Massey et al. 2007b and Ferry et al. 2008), which empirically mimics the complex morphologies of galaxies seen in real data, such as the Hubble Ultra Deep Field (UDF: Beckwith et al. 2006) or *Hubble Space Telescope* (*HST*) COSMOS survey (Scoville et al. 2007). The galaxy morphologies are captured via a shapelet decomposition (Refregier et al. 2003; Refregier & Bacon 2003; Bernstein & Jarvis 2002; Massey & Refregier 2005; Massey et al. 2007a), which also makes it easy to introduce a specified weak gravitational lensing signal, useful to hone shear measurement methods.

The *simage* code is written in IDL and can be downloaded.[6] It also requires the core shapelets package, available from the same location, and certain routines from the NASA Goddard Space Flight Center (GSFC) IDL astronomy user's library. Those required routines are bundled in the *simage* download, but updated versions may be periodically available.[7] In addition, exploiting the full potential of some sections of *simage* requires SExtractor.[8]

This article highlights the full capabilities of the code, its general structure, and a number of possible uses. In § 2, we briefly detail previous applications and associated results, plus the strengths and weaknesses of the utilized shapelet formalism. In § 3, we describe the structure of the software package, and discuss the main modules of the two software packages and their uses. We give a brief summary in § 4.

---

[1] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109.

[2] Northwestern University, Department of Physics and Astronomy, Evanston, IL 60208.

[3] Institute for Astronomy, University of Edinburgh, Royal Observatory Blackford Hill, Edinburgh, EH9 3HJ, UK.

[4] Harvard University, Department of Physics, Cambridge, MA 02138.

[5] California Institute of Technology, Pasadena, CA 91125.

[6] From http://www.astro.caltech.edu/~rjm/shapelets.

[7] From http://idlastro.gsfc.nasa.gov/.

[8] Available from http://astromatic.iap.fr/software/sextractor/.

2. CAPABILITIES AND APPLICATIONS OF *simage*

At the heart of *simage* is the ability of the shapelets method to efficiently and flexibly reconstruct complex galaxy morphologies (Fig. 1. Shapelets are a complete, orthogonal set of basis functions, and a weighted linear combination of these can represent any localized image (Refregier et al. 2003; Bernstein & Jarvis 2002; Massey & Refregier 2005). This is analagous to a Fourier transform, where weighted combinations of sines and cosines can be used to reconstruct nonlocalized images. The mathematical properties of the shapelet basis set make it particularly convenient for astronomical image processing, including quick convolutions with point-spread functions (PSF), pixelization, and operations such as translations, rotations, magnifications, and shears, that can be used to add a known signal into a simulated image (Refregier & Bacon 2003; Massey et al. 2007a). The ability to represent shears in a simple manner allows for the inclusion of distorting shears produced by both the telescope's optical systems and weak gravitational lensing within galaxy clusters. While a Gaussian-based shapelet representation of a galaxy with a particularly strong central peak, or extended wings, is not ideal due to the difficulty of reproducing such features with Gaussian forms, a number of tests to reproduce exponential radial profiles with shapelet basis functions have shown good reproduction of the majority of galaxy shapes with very little bias (Massey et al. 2007b).

The simulated images are capable of being produced in the B, V, i, and z bands, since they are based on a galaxy morphology catalog preconstructed from the Hubble UDF. Hence, the available passbands allowed are the F485W, F606W, F7775W, and F850LP filters of the *HST*.

For a more in-depth description of the method, including a general mathematical introduction to its application to image simulation, the reader is directed to the aforementioned references and Appendix A. In this section, we shall highlight the capabilities such a shapelet formalism provides, along with associated applications and results.
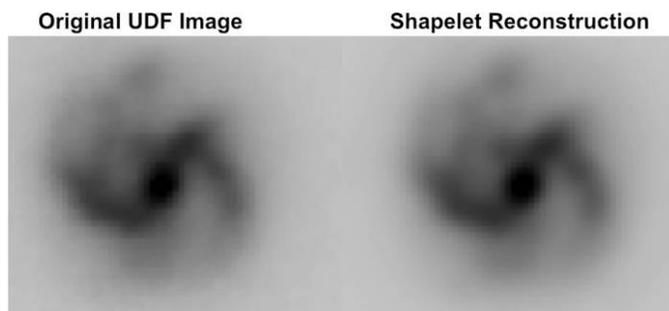

Original UDF Image   **Shapelet Reconstruction**

FIG. 1.—Example of a spiral galaxy from the Hubble Ultra Deep Field, *left,* and one modeled using shapelets, *right*. It can be seen that the model easily captures the major features of the original galaxy (Massey & Refregier 2005).

2.1. Prior Applications: Shear Studies, Data Codec, and Mission Development

Previous applications of shapelets for image simulation, in its *simage* incarnation as presented here and in other forms, have ranged from direct image creation for community shear-analysis studies, data compression investigations, and telescope/instrumental development. We will briefly detail these below.

The Shear TEsting Program (STEP) was a collaborative project which aimed to improve the accuracy and reliability of all weak gravitational lensing measurements in preparation for the next generation of wide-field surveys. STEP was launched in order to test and improve the accuracy and reliability of all these methods through the rigorous testing of shear measurement pipelines, the exchange of data and the sharing of technical and theoretical knowledge within the weak lensing community (Massey et al. 2007b). Here, the shapelets-based *simage* code was used to create image data with incorporated shear fields for analysis in the testing program. Subroutines of the code were also used in the analysis of the resultant data. Schrabback et al. (2009) used images with simulated shear produced by this software to verify and validate their weak lensing code, which they then used to provide the first direct detection of dark energy using weak lensing tomography. Opening up the issue to wider participation, particularly that of computer scientists, the GRavitational lEnsing Accuracy Testing 2008 (GREAT08; Bridle et al. 2008, 2009) again allowed for the application of *simage*'s image simulation capabilities.

In Vanderveld et al. (2010, in preparation), *simage* was utilized to test compression-decompression (codec) algorithms and methods for future visible survey telescopes. This is of vital importance given the vast quantities of data both space- and ground-based survey telescopes are predicted to produce in the coming years (tens of petabytes; e.g., LSST, Ivezic 2007). The roll of *simage* here was to create batches of simulated image data that recreated sizes, morphology, fluxes, and shapes that accurately mimic those we might expect from a visible survey telescope both in orbit and on Earth.

Specific mission development has also been an application of the *simage* routines. In particular, the simulation pipeline has been a critical optimization tool in the optical design and observation strategy for ESA's cosmic visions candidate Euclid (Refregier et al. 2010), the Nasa/DOE Joint Dark Energy Mission (JDEM) concept (e.g., SNAP; Jelinsky 2006; High et al. 2007), and weak lensing balloon missions.

2.2. Example Application

As a demonstration of one of *simage*'s potential uses, we present an example investigation intended to explore telescope survey depth, specifically the relation between the effective number of galaxies observed in a survey sample (n_{eff}), versus pixel scale and exposure time for a mock-up space telescope design (the n_{eff} quantity in this context is the number of galaxies

TABLE 1

EXAMPLE INPUTS WITHIN TELESCOPE.PARAM FOR A MOCK-UP
TELESCOPE DESIGN

Parameter File Input	Value
throughput_ratio	[0.0, 0.0, 1.2, 0.0, 0.0, 0.0]
pixel_scale	$[0.0, 0.0, 0.1, 0.0, 0.0, 0.0]''$ pixel^{-1}
read_noise	5 electrons
collecting_area	1.13 m^2
band_begin	2
band_end	2
exposure_time	100 s
area	6.0 arcmin^2
n_star	1000
n_gal	Default
ee50	0.15$''$

within a given survey area that meet a desired size and signal-to-noise ratio, S/N). A list of possible input telescope and instrument parameters are shown in Table 1. We generate images according to these inputs using a varying the pixel scale between 0.05 to $0.40''$ pixel^{-1} (with constant exposure = 400 s), and the exposure time between 100–800 s (with constant pixel scale $0.1''$ pixel^{-1}). An example of the output as created from a set of images simulated from *simage* is shown in Figure 2.

3. THE SIMULATION PIPELINE

This section will detail the critical internal routines of *simage*, and describe the flow of the simulated image creation.

3.1. Module Overview

The routine simage.pro is the primary program; it utilizes various routines within the pipeline to manufacture a simulated image. Keywords listed in Table 2 can be used to specify the desired telescope and survey characteristics. By default, the image will be produced in the B, V, i, and z bands, based on a galaxy morphology catalog preconstructed from the Hubble UDF. More permanent changes to the telescope and survey characteristics can be fixed in the telescope.param file.

Figure 3 details the pipeline's main processes in the form of a flow chart. We note from the chart that there are three main stages to the pipeline: the multiwavelength catalog generation, the repopulation of the catalog images into a field/resolution governed by the desired telescope parameters, and finally the addition of the various noise components. Other important routines in the pipeline are:

simage_make_shapelet_object.pro—Generates a pixellated image of one simulated galaxy from a given set of shapelet coefficients.

simage_make_analytic_object.pro—Generates an object for the image simulations, using an analytic profile. The size, magnitude, and ellipticity are drawn from a real UDF galaxy template.

num_counts_frac.pro—Calculates the galaxy magnitude distributions normalized to the COSMOS survey data at mid magnitudes (Leauthaud et al. 2007), and to a compilation of other surveys at low and high magnitudes (Metcalfe et al. 2001; see § 3.1.2 for discussion.).

get_telescope_psf*—Reads in the desired PSF FITS file before converting it into shapelet space (* telescope here represents a variety of telescope or survey names included in
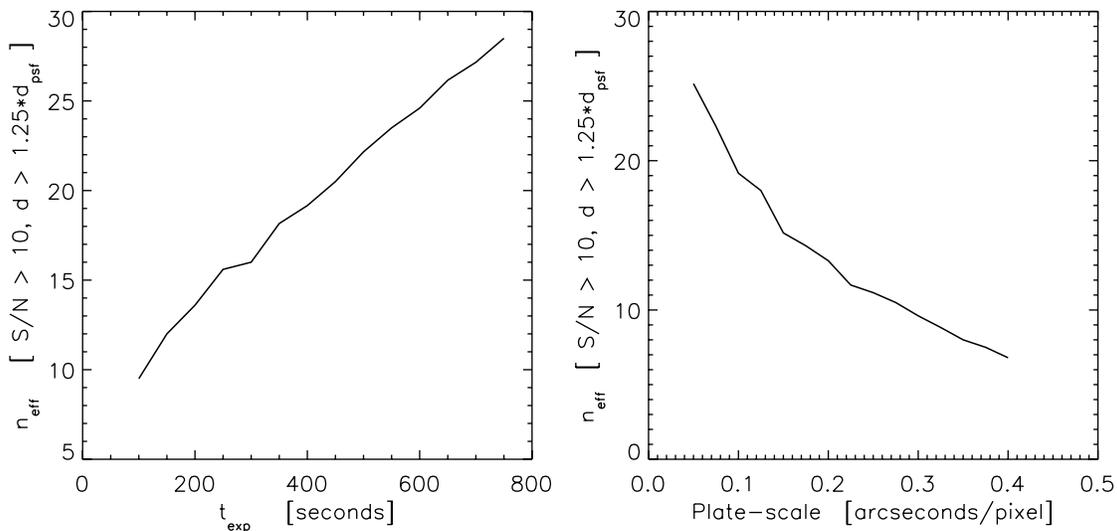

FIG. 2.—*Left:* Plot of $n_{\rm eff}$ galaxy counts vs. exposure time at a pixel scale of 0.1$''$. The $n_{\rm eff}$ quantity in this context is the number of galaxies within a given survey area that meet a particular desired size and S/N. *Right:* Plot of $n_{\rm eff}$ galaxy counts vs. pixel scale at an exposure time of 400 s. These particular plots were generated from images created in the i band (band 2 in *simage*) for a mock-up telescope with a 1.2 m diameter mirror over a 6 arcmin^2 area of the sky; see Table 1.

TABLE 2
DESCRIPTIONS OF THE USER INPUTS FOR TELESCOPE.PARAM

Parameter file input[a]	Description
throughput_ratio	Total system throughputs relative to UDF
sky_level	Skylevel for each band (counts s arcsec^2)
zeropoint	Chosen zeropoint for the images in each band
pixel_scale	The instrument pixel scale ($''$ pixel^{-1})
psf_type	Selects which PSF to use
psf_path	Path to the user's chosen PSF FITS file
read_noise	CCD read noise in number of electrons
background	0, background subtracted; 1, background added
collecting_area	The mirror collecting area (m^2)
band_begin	The band on which to start the simulations
band_end	The band on which to end the simulations
exposure_time	Exposure time (s)
area	The area on the sky to simulate (arcmin^2)
random_seed	A random seed for all random selections
gamma	The user specified weak lensing shear
output_file_pref	Selection of output image file names
n_star	Number of field stars to be added
n_gal	Number of field galaxies
filter_files	Path to user's transition filter files
ee50	The half-light radius of the PSF
fwhm	The full-width at half-maximum of the PSF

[a] These parameter inputs govern telescope design and in turn the resultant output images. Some parameter inputs are degenerate, e.g., if a throughput_ratio is entered, the filter_files path is ignored.

the pipeline, e.g., get_udf_psf, etc.). The PSF is an array of odd dimensions in logarithmic units, by convention.

The desired size of the PSF is quantified in two ways: via the radius that encloses 50% of the PSF energy (EE50 or "half-light radius"), or the full width at half-maximum (FWHM). The EE50 term is more commonly used in optical engineering, whereas FWHM is more commonly used in science applications. Both are available as user inputs and both have particular advantages when creating simulated images. For example, when comparing the EE50 to that of the FWHM, the former is more affected by the tail's profile. For this reason, EE50 is a better measure of how compact something is if you only have one number to describe an object. As a comparison, a Gaussian with a FWHM of 1.77 pixels would have an EE50 of 0.885 pixels, while a typical PSF with the same FWHM of 1.77 pixels would have an EE50 of 1.21 pixels.

3.1.1. Input Catalog Generation

As described in Ferry et al. (2008), the first task is to generate a catalog of galaxy morphologies from real data. Note that galaxy morphologies are already provided from the UDF and, until better data become available, this time-consuming section of the pipeline need not be rerun. However, if desired it is possible to regenerate the UDF catalog, or indeed to generate additional catalogs.
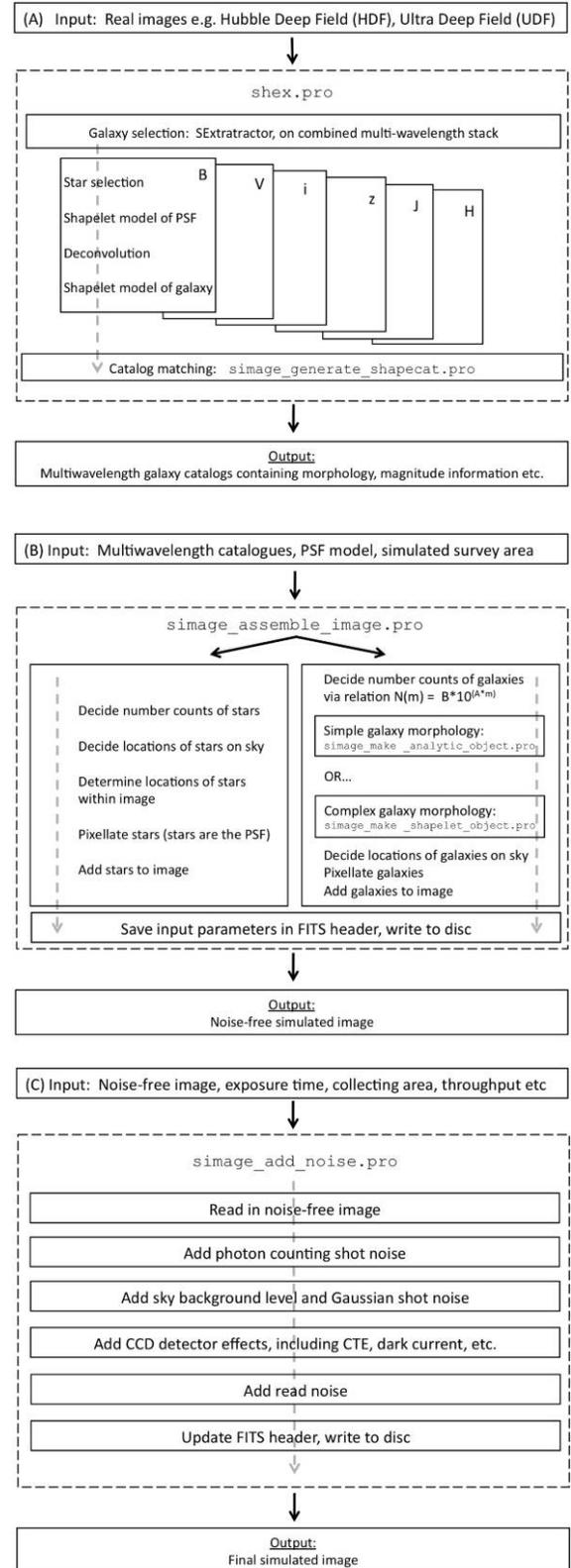

FIG. 3.—Flow chart of the simulation pipeline's key processes beginning with the initial input UDF images to the final output FITS images.

The simulated images are based on UDF data, with photometric redshifts from Coe et al. (2006). The process of getting from real data to simulated images, using modules included in the simulation pipeline software package, is as follows. First, objects in the real data are detected and cataloged using the SExtractor routine on the image using a specified configuration file, config.sex, an example of which can be found in the *analysis* directory of the *simage* package. The cataloged objects are then decomposed into shapelets by running shex.pro. This program takes the real image, the SExtractor catalog, and the desired n_{max} as inputs, and outputs a catalog of shapelet coefficients for the objects in the SExtractor catalog. Here, n_{max} is a measure of the order of shapelet decomposition (higher n_{max} gives a better shape representation). We choose $n_{max} = 20$ for the optical band, this being sufficient to model the *HST* PSF. shex.pro also uses the shapelets focus suite of routines to optimize the n_{max}, β, and centroid parameters, where β is a scaling factor for the size of the PSF. The program also outputs flags, from 0 to 10, to tell the user how well a given object was modeled with shapelets, 0 being good and 10 signaling failure. A list of the flags and their corresponding criteria are shown in Table 3.

There is also an option in shex.pro to remove a specified constant PSF, which can be modeled from the stars in the image. The PSF removal from the UDF catalog is a three step process. Firstly, the *HST* PSF is modeled by selecting stars in the UDF images and decomposing them into a sum of shapelet basis functions. These stars are the best representation of the PSF contained within the image. Secondly, the galaxy objects of an image, once also decomposed into shapelet space, have the star/PSF shapelets subtracted, thus leaving a catalog of galaxies with the *HST* PSF removed. The stars in the newly created catalog are then discarded, since any PSF model that will be introduced will be formed from new simulated stellar images.

TABLE 3
MEANING OF FLAGS OUTPUT FROM SHEX.PRO DETAILING TO
THE USER HOW WELL A GIVEN OBJECT WAS MODELED
WITH SHAPELETS

Flag[a]	Status
0	OK
1	Nearby object
2	Severe overlapping with nearby object
3	Object is near a saturated pixel
4	Object is near a masked region
5	Object is near the edge of the image
6	Object is itself masked out
7	Object has 0 FWHM
8	Too few background pixels around object
9	Object entirely overlapped by neighbors
10	Routine sexcat2pstamp crashed

[a] 0 value implies a successful decomposition, while a 10 signals failure. A FWHM of 0 implies a profile fit to the object failed.

Galaxies then have to be cross-matched across bands of data. This is done using srcor.pro, in the IDL Astronomy Library, to a tolerance of 1 arcsecond. Some galaxies will appear in all bands and some will appear in a subset, but each galaxy is given an ID number and then a master catalog is created that contains the information for each unique galaxy ID, including its position, redshift, and shapelet coefficients in each band. For the UDF, this catalog is stored as a structure called shapecat_total_trim.sav, which should be located in the specified *data* directory alongside the necessary *psf* folder. The core routine, simage.pro, then randomly draws galaxies (in the form of their decomposed shapelet coefficients) from this catalog when producing simulated images.

3.1.2. Noiseless Image Simulation

The pipeline would then proceed as follows: The simulated image is scaled according to the instrument and filter throughput. These are defined by parameters throughput_ratio and filter_files. The first specifies the throughput ratio for each band compared to that of the *HST* Advanced Camera for Surveys (ACS), should it already be known by the user. The second option allows you to calculate this throughput_ratio array by specifying an arbitrary filter curve. This filter_files lists the total desired instrumental throughput for each band, in the format of wavelength (Å) versus throughput, where 1.0 represents 100% transmission. The filter curve is then integrated and compared against the *HST*/ACS filter curve to generate the correctly normalized array, throughput_ratio. In this way, it is possible to use UDF images but normalized for the throughput of any given telescope design. However, note that while the integral of the throughput is considered, its shape is not. For example, the shape of galaxies is therefore not changed by a transmission curve that peaks redward of the *HST*/ACS filters and should therefore enhance the bulges of galaxies. The pipeline then reads in the specified PSF file, which remains constant throughout the simulation.

The simage.pro routine will then pass to simage_assemble_image.pro which reads in the UDF shapelet catalog and populates the image with either n_gal number of galaxies, or a predefined default that is calibrated to existing observational data (the *HST* COSMOS survey; Scoville et al. 2007). The distribution of the galaxies in the image is random, and there is no galaxy clustering (high–galactic-latitude, "empty-field" observations). At this point, any specified weak lensing shear, represented by the two-dimensional array $[\gamma_1, \gamma_2]$, is added to each galaxy, i.e., a constant value across the field (see Appendix A). The pixel scale of the shapelet catalog galaxies are adjusted to the specified value at this stage. A similar process is performed to populate the image with field stars, each of which are represented by the PSF. Here the subroutine simage_star_magnitude_distribution.pro generates a random flux level for stars in the image. The stars follow

the stellar luminosity function measured at the galactic poles and tabulated in Allen (2000).

We note that the pipeline also has the ability to create a simulated image containing objects with analytic (e.g., de Vaucouleurs) profiles with the same size, magnitude and ellipticity distributions as the UDF shapelet catalog. The shapelet coeficients are not used for any purpose other than the determination of these distributions. Hence the pipeline can also use a non-shapelet method to create simulations, which can themselves be used to test shapelet-based shape-measurement techniques if desired.

The output noiseless image is written in units of photons or counts per second, along with a mock SExtractor output file with all the objects' known input positions, sizes, magnitudes, ellipticities and star/galaxy classifications. All files are output to the *Data* directory.

Although the galaxy catalogs are created from the UDF, the magnitude distribution of the resultant simulated images is normalized to a variety of existing galaxy surveys rather than just drawing randomly from the UDF galaxy magnitudes. Since the decomposed shapelet objects have color representative of real galaxies, this approach can apply to all bands available to the simulation pipeline (i.e., B, V, i, and z). The routine num_counts_frac.pro utilizes the number (N) counts relation

$$N = B * 10^{(A*m)} \tag{1}$$

where A and B are normalization factors, and m is the galaxy magnitude. The form of this expression and the values of the normalization factors A and B are taken from existing COSMOS survey analysis between apparent magnitudes 21 and 26 (Leauthaud et al. 2007). COSMOS is the widest *HST* survey and as such is least affected by cosmic variance/sample size. It is also the closest data to the future space surveys that *simage* aims to model. Below apparent magnitude 21 and above apparent magnitude 26, the number counts are fit to various survey sources as compiled in Metcalfe et al. (2001). The form is continuous at these transition points and results in a realistic number count relation for the simulated images at low, mid, and high magnitudes.

In Figure 4 we display a comparison of galaxy sizes in the HST-COSMOS survey and those from an *simage* simulated COSMOS field. We see that the normalization of *simage* to the COSMOS counts discussed above allows the pipeline to obtain a very similar size distribution. We note that there are differences at the lower size limit due to *simage* reproducing fewer very small galaxies in the simulation. This is because the pipeline does not decompose the smallest galaxies into shapelets since some of them are noise (or noisy) and as such do not end up in the shapelets catalog.

3.1.3. Addition of Noise

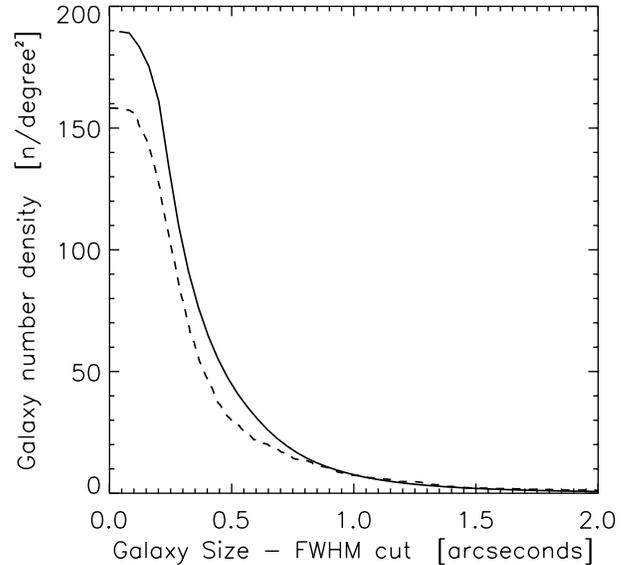The routine simage_add_noise.pro reads in the FITS format noise-free image, and writes out a noisy image, plus an

Fig. 4.—Comparison of galaxy number density vs. galaxy size for both real HST-COSMOS survey data (*solid line*; Leauthaud et al. 2007) and *simage* simulated COSMOS data (*dashed line*). The galaxy size is measured via the SExtractor *FWHM_IMAGE*. The x-axis scale is the FWHM cut size, i.e., galaxies that have greater than a given size. Simulated data used COSMOS survey parameters of $0.03''$ pixel^{-1} and an exposure time of 2000 s.

inverse variance weight map. This is very fast to run, and is intentionally kept separate from the previous sections because a common task is to investigate the effect of changing the survey exposure time. In this case, the same noise-free image can be used, and simage_add_noise.pro run multiple times in isolation, with different input parameters. Specific noise features and detector effects that can be added to an image post-creation, e.g., dark current, are included within the simage_add_noise.pro routine itself and are activated by selecting the corresponding flags when the routine is initially called.

For convenience, the noise model is calculated in two separate components: shot noise on astrophysical sources, and on the sky background. In both cases, a random distribution of uncorrelated pixel values is drawn from a Gaussian with width equal to the square root of the counts in each pixel. The sky background level is estimated by default from that in the UDF, but can also be specified in telescope.param, in units of counts s arcsec^{-2}. By default, the constant sky background level is then subtracted, although this behavior can be turned off via the *background* keyword (see Table 2).

Additional options include the ability to add read noise, to correlate the background noise to cheaply mimic the effects of DRIZZLE, to truncate saturated pixels, and to truncate at zero any nonphysical, slightly negative pixel values (arising from noise or modeling problems in the original UDF). None of these are enabled by default. To mimic the effects of DRIZZLE, the image is convolved with a kernel similar to the drizzle drop

kernel and the original pixel square. This has the effect of correlating adjacent pixels in the image, and is most noticeable in the background noise in blank areas well away from sources.

4. SUMMARY

We have introduced an IDL simulation pipeline, *simage*, that creates mock deep field survey images by drawing from a catalog of Hubble Ultra Deep Field galaxies. Each galaxy in the catalog is decomposed into a set of analytic shapelet basis functions which can completely describe their morphological properties in a simple manner. We have shown how this catalog can then be used to populate a field of any given size and resolution depending on the user's requirements. The pipeline allows the user complete control over parameters such as exposure time, PSF type, mirror size, pixel scale, field star density, and noise, and simulates fields in the B, V, i, and z bands.

The code also has the ability to introduce a weak lensing signal into the data, allowing the output to be used for studies into weak lensing reconstruction analysis. It is envisioned that the code will be used as a tool for research, instrumental development, and teaching. It is available to download as a self contained package of IDL modules.

APPENDIX A

SHAPELETS FORMALISM

Shapelets come in two flavors: Cartesian shapelets are separable in x and y, and polar shapelets in r and θ. There is a one-to-one mapping between the two, so without loss of generality, we shall adopt whichever has the more convenient symmetries for the task at hand. The polar shapelet basis functions

$$\chi_{n,m}(r,\theta;\beta) = \frac{(-1)^{\frac{n-|m|}{2}}}{\beta^{|m|+1}} \left[\frac{(\frac{n-|m|}{2})!}{\pi(\frac{n+|m|}{2})!} \right]^{\frac{1}{2}}$$
$$\times r^{|m|} L_{\frac{n-|m|}{2}}^{|m|} \left(\frac{r^2}{\beta^2} \right) e^{\frac{-r^2}{2\beta^2}} e^{-im\theta}, \qquad (A1)$$

where $L_p^q(x)$ are the Laguerre polynomials, have an overall scale size β and are parameterized by two integers, n and m, which are the number of oscillations in the radial and tangential directions. The basis functions are calculated using `shapelets_chi.pro`. Using `shapelets_decomp.pro`, a galaxy (or star) image $I(r,\theta)$ can then be decomposed into (complex) "shapelet coefficients" $f_{n,m}$

$$f_{n,m} = \iint_{\mathbb{R}} I(r,\theta)\chi_{n,m}(r,\theta;\beta)rdrd\theta, \qquad (A2)$$

so that the (wholly real) image can be reconstructed, using `shapelets_recomp.pro` as

$$I(r,\theta) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} f_{n,m}\chi_{n,m}(r,\theta;\beta). \qquad (A3)$$

In practice, it is necessary to truncate the expansion at some maximum value of n. Figure 1 shows an example galaxy image and its reconstructed counterpart using shapelets up to order $n_{\max} = 20$. It can be seen that the model easily captures the major features of the original galaxy.

In shapelet representation, convolution between two images (such as a galaxy and a telescope's PSF) is simply a matrix multiplication of their $f_{n,m}$ coefficient arrays (Refregier et al. 2003). In the code, this is implemented via `shapelets_convolve.pro`. It is also possible to perform a deconvolution by inverting the PSF matrix; this is incorporated within `shapelets_decomp.pro`.

While previous operations were performed in polar shapelets since the functions are separable in r and θ (rendering many operations more intuitive), pixellization can be performed most easily by switching to Cartesian shapelets, then switching back. A closed form for the integrals of Cartesian shapelet basis functions over rectangular pixels is given in § 4.3 of Massey & Refregier (2005), and is enabled by default in `shapelets_chi.pro`.

We shall use several transformations of the galaxy images, first to randomize their appearance in the final simulation, and then to impose a gravitational lensing signal. In shapelet space, galaxies can be easily rotated by adjusting the phase of their (complex) coefficients $f_{n,m}$, or reflected in the x-axis by taking their complex conjugates. A weak gravitational lensing shear signal γ can be applied to first order by mixing adjacent coefficients according to the mixing matrix

$$\widehat{1} + \widehat{S}(\gamma): f_{n,m} \rightarrow f'_{n,m} = f_{n,m}$$
$$+ \frac{\gamma}{4} \{ \sqrt{(n+m)(n+m-2)} f_{n-2,m-2}$$
$$- \sqrt{(n-m+2)(n-m+4)} f_{n+2,m-2} \}$$
$$+ \frac{\gamma^*}{4} \{ \sqrt{(n-m)(n-m-2)} f_{n-2,m+2}$$
$$- \sqrt{(n+m+2)(n+m+4)} f_{n+2,m+2} \} \qquad (A4)$$

as described in § 2.3 of Massey et al.(2007a), which also provides similar operations for flexion. In equation (A4), $\widehat{1}$ corresponds to the identity operator, while $\widehat{S}$ corresponds the to shear operator. Routines to implement such operations in practice are located in the `shapelets/operations/` subdirectory.

This prescription for shear is only accurate to order γ. This will be insufficiently accurate for very high precision work, or if the gravitational lensing signal is particularly large. A new innovation for *simage* is that this transformation can now be generalized to include higher order γ^2, γ^3, etc., terms. This is achieved mathematically by exponentiating the operation (Bernstein & Jarvis 2002). For a practical implementation to fourth order, note that the first four terms in an exponential expansion

$$\widehat{1} + \widehat{S} + \frac{\widehat{S}^2}{2!} + \frac{\widehat{S}^3}{3!} + \frac{\widehat{S}^4}{4!} = \frac{3}{8} + \frac{\widehat{1} + \widehat{S}}{3} + \frac{(\widehat{1} + \widehat{S})^2}{4} + \frac{(\widehat{1} + \widehat{S})^4}{24},$$
(A5)

where $\widehat{S}$ here is the shear operator but could equally be replaced by any other. To perform this on a shapelet model we simply need to apply the linear mapping in equation (A4) 4 times, recording the new coefficients $f'_{n,m}$ at each stage, and add them to the original coefficients in the ratio $\frac{3}{8} : \frac{1}{3} : \frac{1}{4} : 0 : \frac{1}{24}$. This behavior is controlled via the *order* keyword in `shapelets_shear.pro`, and is set to 4 by default.

Note that, as discussed in Bernstein & Jarvis (2002), there is a somewhat arbitrary choice for these higher order terms, which can be changed depending on the required definition of shear. The expansion in equation (A5) changes an intrinsically circular source into an ellipse with major and minor axes a and b via a distortion $\delta \equiv (a^2 - b^2)/(a^2 + b^2)$. A "conformal shear," $\nu = \text{arctanh}(\delta)$, produces a slightly different ratio of major and minor axes, but can be achieved by simply adjusting the input shear. A fourth-order implementation of a conformal shear in shapelet space perfectly matches the real-space transformation of highly oversampled images within a computer's numerical precision (Rowe, B., 2008, private communication). It is therefore not just faster, but should be accurate within 1% for shears up to $\gamma \approx 0.47$ (Massey & Goldberg 2008). For a typical cosmological gravitational lensing signal of a few percent, applying only a first order shapelet-based shear yields pixel values in the final image that are incorrect at a level of approximately one part in 10^{-3}, and changing from δ to ν yields differences of around one part in 10^{-5}. Because of this, the *simage* pipeline routines, such as `simage_make_analytic_object.pro`, use the conformal shear, ν, when called to include a shear signal.

REFERENCES

Allen 2000, Allen's Astrophysical Quantities, ed. A., Cox (4th ed.; London, UK: The Athlone Press)

Beckwith, S. V. W., et al. 2006, AJ, 132, 1729

Bertin, E. 2009, Mem. Soc. Astron. Italiana, 80, 422

Bernstein, G., & Jarvis, M. 2002, AJ, 123, 583

Bridle, S., et al. 2009, preprint (arXiv:0908.0945)

———. 2008, AnApS, 3, 6

Coe, D., Bentez, N., Sánchez, S., Jee, M., Bouwens, R., & Ford, H. 2006, AJ, 132, 926

Erben, T., van Waerbeke, L., Bertin, E., Mellier, Y., & Schneider, P. 2001, A&A, 366, 717

Ferry, M., Rhodes, J., Massey, R., White, M., Coe, D., & Mobasher, B. 2008, Ann. Phys., 30, 65

High, F. W., Rhodes, J., Massey, R., & Ellis, R. 2007, PASP, 119, 1295

Ivezic, Z. 2007, A&AS, 210, 6605

Jelinsky, P. 2006, A&AS, 209, 9809

Leauthaud, A., et al. 2007, ApJS, 172, 219

Refregier, A. 2003, MNRAS, 338, 35

Refregier, A., & Bacon, D. 2003, MNRAS, 338, 48

Refregier, A., & Euclid Imaging Consortium, et al. 2010, preprint (arXiv:1001.0061)

Rhodes, J., Refregier, A., & Groth, E. J. 2000, ApJ, 536, 79

Massey, R., & Goldberg, D. 2008, ApJ, 673, 111

Massey, R., Refregier, A., Conselice, C., & Bacon, D. 2004, MNRAS, 348, 214

Massey, R., & Refregier, A. 2005, MNRAS, 363, 197

Massey, R., Rowe, B., Refregier, A., Bacon, D., & Berge, J. 2007a, MNRAS, 380, 229

Massey, R., et al. 2007b, MNRAS, 376, 13

Metcalfe, N., Shanks, T., Campos, A., McCracken, H. J., & Fong, R. 2001, MNRAS, 323, 795

Schrabback, T., et al. 2009, preprint (arXiv:0911.0053)

Scoville, N., et al. 2007, ApJS, 172, 1