

# Report on Winter 2014 Production: Image Differencing

April 4, 2014

The goals of this Winter 2014 (W14) task are to investigate the effects of differential chromatic refraction (DCR) on the rate of false positives in image differences. We quantify this using a suite of image simulations that contain no intrinsic variability, and run the images through the LSST Data Management image subtraction pipeline, which includes detection and measurement of false positives (both positive-going and negative-going) in the differences. As demonstrated in a similar analysis in Winter 2013 (W13), we are able to achieve a false detection rate consistent with random Gaussian fluctuations in the background in the absence of DCR.

The underlying image simulation software (`phoSim`) and the DM stack have evolved since our W13 work, `phoSim` from v3.2.2 to v3.3.2 and DM from v7.2 to v7.3. Accordingly, our first subtask was to reproduce the results of W13, which involve a suite of 15s *i*-band images taken at a zenith angle of 20.2° and in seeings of 0.66, 0.8, and 1.2 arcseconds. These were differenced against a 300s *i*-band image generated at the same airmass, with seeing 0.8''. The focus of this analysis was the image-vs-template seeing dependence of the rate of false positives, and how pre-filtering of the science `Exposures` with their `Psfs` affects this rate. We limited our reanalysis to the central sensor of central raft 2, 2, and quantitatively (but not exactly) recover the results of W13. The second subtask involved the generation of *g*-band and *r*-band images in the same observing configuration, to verify there is no wavelength dependence to our results. We find that our results are quantitatively similar to those of the first subtask. We do not see substantial issues with DCR in these data, as expected.

Our third subtask involved creating a similar suite of *gri*-band images, but at 5 airmasses throughout a single night, using a star catalog with 3 discrete SEDs. The data include 2 observations before meridian crossing at airmasses 1.55 and 1.16, one observation near zenith, and then 2 observations after the meridian crossing at airmasses 1.16 and 1.55. Observations at the same airmass, but before and after meridian crossing, have different parallactic angles, and thus different directions of DCR. This directly tested whether or not the parallactic angle needs to be a consideration when designing templates for image subtraction during LSST operations. We differenced each of 5 per-airmass templates against each of the 15 science images (binned 5 in airmass and 3 in seeing), and did this in each of the 3 passbands. We find that the difference images where the parallactic angles are aligned yield false positives consistent with the rates seen above, at all airmasses. However, when differencing images taken at one parallactic angle with a template taken at another, we find a significantly enhanced rate of false positives. This rate is both passband and seeing dependent, in the sense that the false positives are more enhanced in the *g*-band than in the *i*-band, and higher in the better seeing data than in the worse seeing data. We find that DCR offsets of 1–2 mas are sufficient to lead to dipole rates similar to those expected from the background fluctuations. This suggests that parallactic angle differences between science and template images of 20–30 degrees in the *g*-band, and ∼ 90 degrees in the *r*-band, will lead to significant false positive rates. A fourth subtask – using a more realistic mixture of stars and galaxies in the input catalog – was deferred to later analysis, as this is not central to resolving the DCR issue.

We validated that the amplitudes and orientations of DCR “dipoles” are similar both to theory and to the effects designed into the simulations. We also find that joint-Psf dipole measurements of DiaSources tend to overestimate the amplitude of the effect, with a dependence on seeing. Finally, we provide extensive documentation on the generation and data reduction of image simulations, how to design DCR into the data, and how to interpret the results.

# Contents

<b>1 Production Scope and Goals</b>	<b>3</b>
1.1 Subtask 1: Reproduce W13 Results Using Current PhoSim . . . . .	3
1.2 Subtask 2: Simulate Starfield using Multiple SEDs at a Single Airmass . . . . .	4
1.3 Subtask 3: Simulate Starfield using Multiple SEDs at Multiple Airmasses . . . . .	4
1.4 Subtask 4: Include Realistic Mix of Stars and Galaxies . . . . .	5
<b>2 Review of Wavelength Dependent Refraction</b>	<b>5</b>
<b>3 Results: Subtask 1</b>	<b>7</b>
<b>4 Results: Subtask 2</b>	<b>8</b>
<b>5 Results: Subtask 3</b>	<b>10</b>
5.1 Dipole Orientation and Amplitude . . . . .	10
5.2 Dipole Rate . . . . .	13
<b>6 Implications for Image Subtraction Templates</b>	<b>13</b>
<b>7 Thoughts on the DCR Issue</b>	<b>14</b>
<b>8 Conclusions</b>	<b>15</b>
<b>Appendices</b>	<b>25</b>
<b>A Ticket Work</b>	<b>25</b>
A.1 Ticket #3128 . . . . .	25
A.2 Ticket #3143 . . . . .	25
A.3 Ticket #3161 . . . . .	25
<b>B Use of PhoSim</b>	<b>25</b>
B.1 Setting up PhoSim . . . . .	26
B.2 Running <code>phosim.py</code> . . . . .	26
B.3 Creating Astrometry Index Files . . . . .	27
B.4 Running <code>processEimage.py</code> . . . . .	28
B.5 Running <code>imageDifferenceWinter2013.py</code> . . . . .	28
<b>C Generation of Instance Catalogs for W14</b>	<b>28</b>
C.1 Subtask 1 . . . . .	28
C.2 Subtask 2 . . . . .	29
C.3 Subtask 3 . . . . .	29
<b>D Refraction and DCR Overlays</b>	<b>30</b>

# 1 Production Scope and Goals

The primary goal of this W14 production was to investigate the effects of DCR on the image differencing process, with the expectation that this informs the requirements on image differencing templates. The current DM baseline is that LSST will have up to 9 templates per passband per region of sky, binned 3x3 in airmass and seeing. The airmass bins are currently segmented only on the zenith distance of the observation. However, this design does not account for the angle that refraction displaces objects within the image, which will be different for every observation, and which ideally requires a per-image (in orientation) and per-object (in amplitude) correction to properly compensate for it.

To examine these effects, we designed a staged analysis where each successive stage builds on the results of the previous stage, but adds an additional layer of complexity. Starting with the base input catalog and simulation configuration used in the previous W13 analysis, our second subtask included multi-passband sims, while the third subtask additionally included multi-airmass and multi-parallactic angle sims. This final stage was sufficient to characterize the effects of DCR on the false positive rate. Technical details on each of the subtasks are provided in Appendix C.

We generated images in three passbands corresponding to the LSST  $g$ -band,  $r$ -band, and  $i$ -band filters. Fifteen-second “snaps” were generated at 3 discrete seeings: 0.6”, 0.88”, 1.2”. One accompanying image subtraction template was also generated (as opposed to assembled via coaddition) with an effective 300 second exposure time and a seeing of 0.88”. All 4 sets of images were generated for each airmass-filter combination.

After generation we ran the simulated **eImages** through single frame measurement (SFM) using the script `processEimage.py`. Note that we did *not* use the raw per-amplifier simulated images, instead choosing to operate on the **eImages** because the former would have required the generation of calibration products and instrument signature removal (ISR). Had we run in this mode, SFM would have proceeded using the script `processCcd.py`.

We next ran the images through image differencing, using the 300s image as the image subtraction template. For this reason, the derived class `Winter2013ImageDifferenceTask` (so-called as it was initially designed for the W13 production cycle) was used, as this includes a specific flag to use a visit as the template, instead of an extraction from the coadd repository. We next ran detection and measurement on the difference image to extract the **DiaSources** from the image. We looked at detections of both positive and negative polarity, and at a detection threshold of 5-sigma. Dipoles were identified and measured, and their separation and orientation compared to what was expected from theory, and what was expected based on the inputs to `phoSim`. The numbers of false positives were recorded as a function of filter, image seeing, image airmass, and (where applicable) template airmass. In the following, we reference the 3 bins in seeing as **Seeings 1,2,3** for seeing = 0.6”, 0.88”, 1.2”, respectively, and the 5 visits at different airmasses as **Visits A,B,C,D,E**. We briefly summarize each of the subtasks below.

## 1.1 Subtask 1: Reproduce W13 Results Using Current PhoSim

To ensure that the results of W14 may be compared to the results from W13, we first validated that the `phoSim` software and DM stack (and their interaction) had not significantly changed in the interim. We note that W13 used `phoSim` version v3.2.2 and DM stack v7\_2; W14 uses v3.3.2 and v7.3, respectively. Starting with the W13 `phoSim` configuration, we produced simulated images at a single airmass (1.07) and in a single passband ( $i$ ). Science images were simulated at each of the 3 seeing values described above, plus an image subtraction template image. We simulated images only for the central sensor 1,1 of the central raft 2,2. We performed image differencing,

source detection and measurement, and validated that the average numbers of false positives were consistent with the results of W13. We also analyzed the actual W13 data using the current version of the DM pipeline, and validated that the total numbers of false positives were similar to the W13 analysis, with minor differences. Results of these analyses are described in Section 3 below.

The `phoSim` configuration files for this subtask are contained in subdirectory `sims3/` of this repository. Work on this subtask resulted in Ticket #3143, described in Appendix A.2 below.

## 1.2 Subtask 2: Simulate Starfield using Multiple SEDs at a Single Airmass

Starting with the input trim files to subtask 1 above, we replaced the SEDs used for the starfield with 3 discrete SEDs chosen to span a range in color. We chose a G0V star to be our reference source, and replaced 80% of the source SEDs with those of SED `km20_6000.fits_g30_6020.gz`. When performing the image differencing, *only* these objects were used for the creation of the Psf-matching kernel. This effectively made this set of sources the reference for the differential refraction. We then replaced 10% of the SEDs with those of a “blue” source – spectrum `kp01_9750.fits_g45_9830.gz` representing an A0V star – and 10% of the SEDs with those of a “red” source – spectrum `m2.0Full.dat.gz` representing an M2.0V star. We modified the brightnesses of the sources in the *i*-band trim file to have approximately the same S/N as the original sources, given the color of each star. We then re-ran the image differencing analysis, and validated that we were able to difference the *i*-band data to a similar quality as in subtask 1 above. The `phoSim` configuration files for this are contained in subdirectory `sims5/` of this repository.

We further generated sets of simulations in the *g*-band and *r*-band using the exact same set of trim files except for the requested `Opsim_filter`. Analysis of these images showed no color-dependence to our results. However, because we did not modify the input brightness distributions for the stars to have similar S/N in each passband, the results are somewhat compromised due to the different signal-to-noise distributions. However, we have not seen S/N dependence in prior analyses, so do not expect this to substantially modify our conclusions. Results of these analyses are described in Section 4 below. The `phoSim` configuration files for this are contained in subdirectory `sims5gr/` of this repository.

Work on this subtask resulted in Ticket #3128, described in Appendix A.2 below.

## 1.3 Subtask 3: Simulate Starfield using Multiple SEDs at Multiple Airmasses

Starting with the `sims5/` trim files referenced above, we further modified the simulations by first shifting the star field to pass through zenith on the night of simulation. This required a shift in declination of approximately 20°, applied to the coordinates of each star and the requested boresight pointing of the simulations. We next evaluated the times that this starfield would pass through 5 discrete zenith angles: 50°, 30°, 0°, 30°, 50°, corresponding to visits A, B, C, D, and E below. This sequence tracks the star field through a single night. The salient configuration parameters of each visit are given in Table 1. We note that 50° is the approximate airmass cutoff of the LSST Universal Cadence. We further modified the brightnesses of each star in each filter’s trim files such that each object was rendered at approximately the same S/N in each filter.

To examine how a mismatch between the effects of DCR in the template and science image affects the false positive rate, we performed each permutation of differencing the template visit ABCDE with science visit ABCDE. The “on diagonal” components (AA, BB, etc) reflect an exact match between the airmass and parallactic angle of the template and science image. Combinations AE and BD (as well as EA and DB) reflect a match between the airmass of the observations, but a mismatch of parallactic angle. All other permutations reflect a mismatch between both airmass and parallactic

Subtask 3: PhoSim Visit Configuration					
Visit	MJD	rotSkyPos	rotTelPos	Altitude	Airmass
A	51130.111719	251.0689567	145.7139810	40.1	1.55
B	51130.176302	251.0689567	152.2708405	59.9	1.16
C	51130.272830	251.0689567	341.0975816	89.9	1.00
D	51130.272830	251.0689567	349.8619334	59.9	1.16
E	51130.433247	251.0689567	356.4181128	40.1	1.55

Table 1: Summary of the configuration parameters that define each subtask 3 visit ABCDE. Observations were designed to follow a single star field through zenith on a single night, and observed twice before and twice after crossing the meridian, as well as at zenith. Each visit was simulated using the same random seed (153555399), in 3 filters ( $g$ ,  $r$ ,  $i$ ), and under 4 observing conditions. These correspond to the template image (`Opsim_rawseeing` = 0.878") and `SIM_VISTIME` = 300) and then 3 bins in seeing for the science image corresponding to `Opsim_rawseeing` = 0.6", 0.878", and 1.2" for seeing bins 1, 2, 3, respectively. All science images were simulated with `SIM_VISTIME` = 15. Overall, this yielded  $5 \times 3 \times 4 = 60$  runs of `phoSim`, which are encoded as `visitId=X00Y00Z` where X represents the filterId  $gri=123$ , Y represents the visit ABCDE= 01234, and Z represents the seeing: 0 for the template and 123 for the science images. All 9 CCDs within `raft=2,2` were simulated.

angle of the images. We performed differences of all 5 template visits with all 5 science image visits, for all 3 filters and all 3 seeing values. Results of these analyses are described in Section 5 below.

The `phoSim` configuration files for this are contained in subdirectory `sims8/` of this repository. Work on this subtask resulted in Ticket #3161, described in Appendix A.3 below.

#### 1.4 Subtask 4: Include Realistic Mix of Stars and Galaxies

This subtask was not undertaken, and is deferred to a future analysis.

## 2 Review of Wavelength Dependent Refraction

We first review the expected signature of differential chromatic refraction in the image simulations. We use the formalism of Filippenko (1982) throughout.

The wavelength-dependent index of refraction of the atmosphere,  $n(\lambda)$ , may be represented as

$$n_0(\lambda) - 1 = 10^{-6} \times \left[ 64.328 + \frac{29498.1}{146 - \lambda} + \frac{255.4}{41 - \lambda} \right] \quad (1)$$

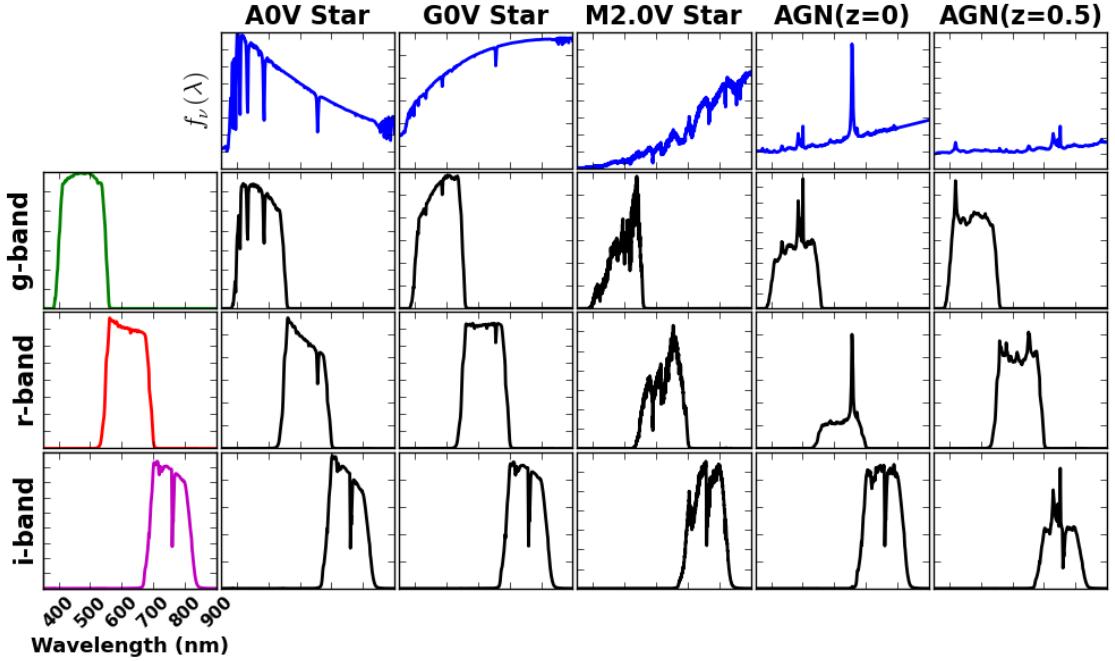
where  $\lambda$  is the wavelength in microns. Temperature, pressure, and water vapor corrections may be expressed as:

$$C1(P, T) = P \times \frac{1 + (1.049 - 0.0157 T) \times 10^{-6} P}{720.883 * (1 + 0.003661 T)} \quad (2)$$

$$C2(f, T) = 10^{-6} \times f \times \frac{0.0624 - 0.000680/\lambda^2}{1 + 0.003661 * T} \quad (3)$$

$$n(\lambda) - 1 = (n_0(\lambda) - 1) * C1 - C2 \quad (4)$$

where  $P$  is the ambient pressure in mm of mercury,  $T$  is the temperature in Celsius, and  $f$  is the water vapor pressure in mm of mercury. The index of refraction is larger for shorter wavelength



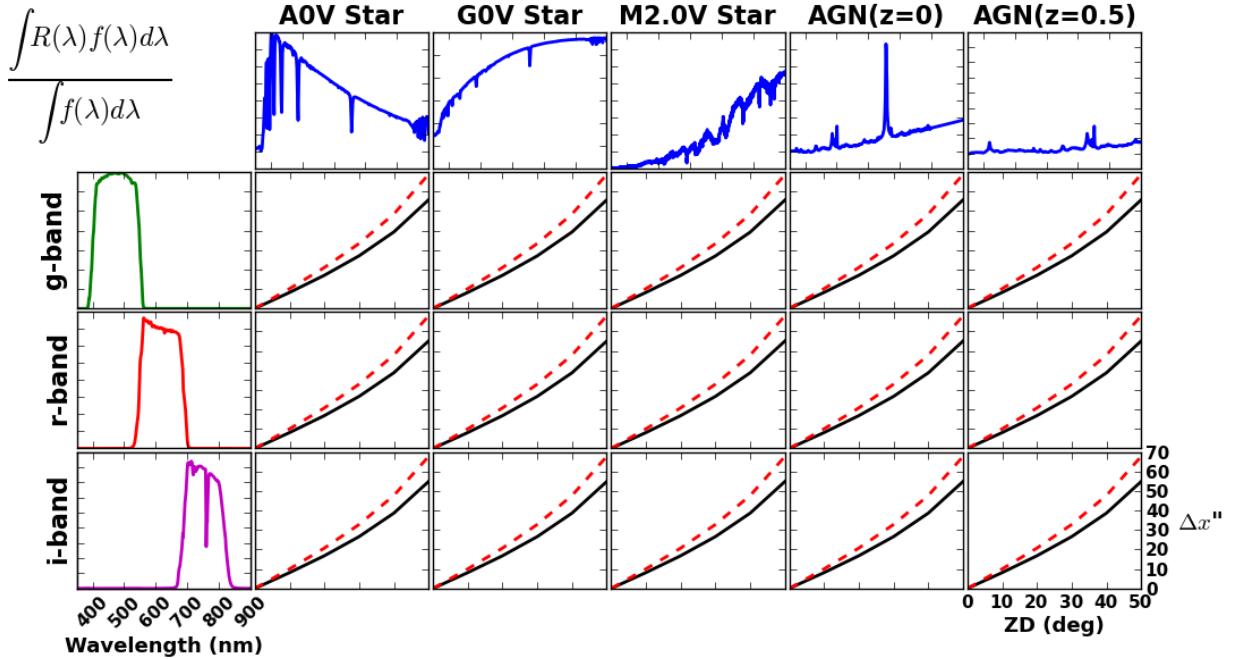
**Figure 1: Effective Spectral Energy Distributions :** The effective spectra of 5 reference objects – a “blue” AOV, a reference GOV, and a “red” M2.0V star, along with a QSO at redshift  $z = 0$  and  $z = 0.5$  – filtered through 3 transmission profiles corresponding to the LSST  $g$ ,  $r$ , and  $i$ -bands at 1.2 airmasses. The top row shows the input spectral energy distribution  $f_\nu(\lambda)$ , while the leftmost column shows the LSST filter transmission profile in units of the normalized system response  $\phi$ . The inner row/column figures show the effective spectrum of each SED (along columns) when multiplied through the respective filter (along rows). In all subpanels, the x-axis is wavelength. The AOV, GOV, and M2.0V spectra correspond to CAT\_SHARE\_DATA files `kp01_9750.fits_g45_9830.gz`, `km20_6000.fits_g30_6020.gz`, and `m2.0Full.dat.gz` respectively, and were used as the SEDs of the stars in the W14 image simulations. This figure may be recreated using the script `python/DCR.py`.

light, meaning that photons from the blue end of the spectrum are refracted more than ones from the red. This effect is dependent on the zenith distance  $Z$ , such that:

$$R(\lambda, Z) = \frac{n(\lambda)^2 - 1}{2 n(\lambda)^2} \tan(Z) \quad (5)$$

where  $R$  is the deflection amplitude in radians. This deflection is expressed along the direction of increasing altitude, such that blue sources will appear higher in the sky, compared to red sources, when seen through the atmosphere. What this means in practice is that there will be a SED-dependent, per-object deflection whose amplitude depends on the source airmass and color, and whose orientation depends on the angle towards zenith in the image.

We examined the theoretical amplitude of this effect using discrete spectra from the phoSim source catalog. We used the same spectra when generating the images for the pixel-level analysis, meaning we expect direct agreement between this theoretical analysis and the expression of DCR in the sims. In Figure 1, we show along the top the spectra, in units of  $f_\nu(\lambda)$ , of 5 sources. Left to right, these represent a “blue” AOV star, a reference GOV star, a red M2.0V star, and an active

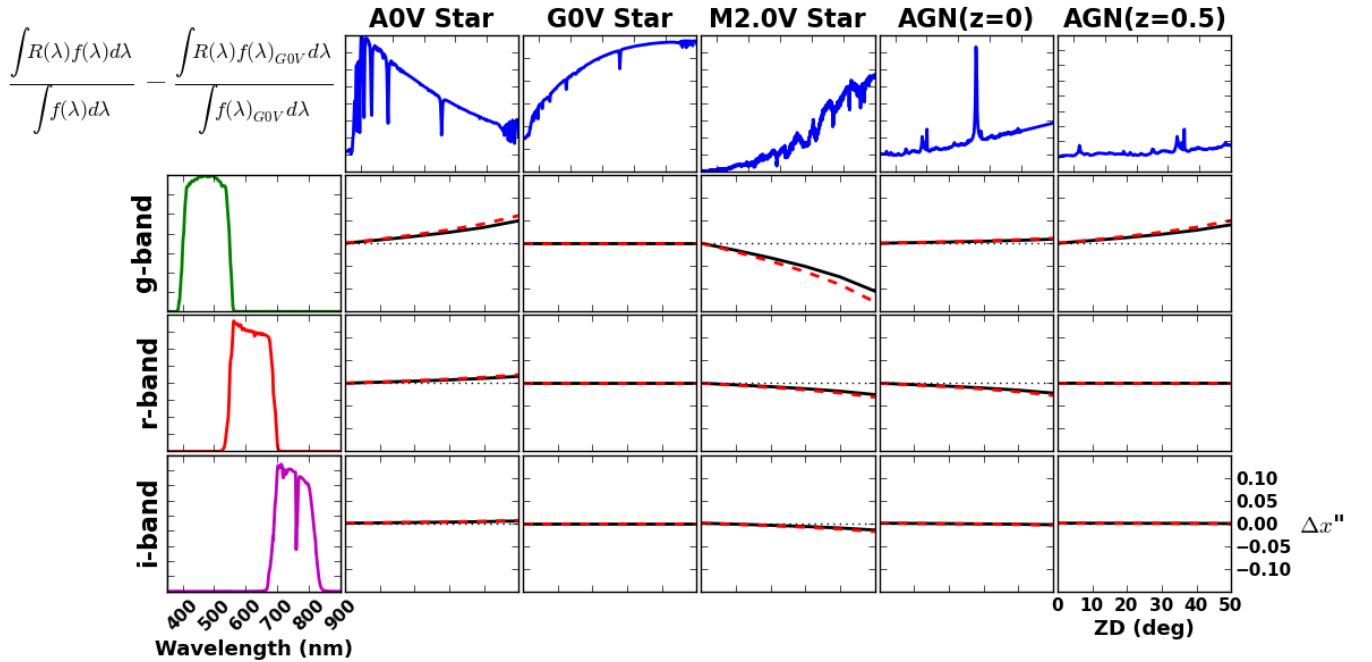


**Figure 2: Refraction Amplitude vs. Filter and Spectral Energy Distribution:** The flux-weighted amplitude of refraction (in arcseconds) for each of the filtered SEDs in Figure 1, as a function of zenith distance in degrees along the x-axis. The solid black line is the nominal result from Eqn 2, while the dashed red line ignores the corrections for temperature and pressure (Eqn 1). Note the maximum amplitude of refraction reaches nearly 1 arcminute. This figure may be recreated using the script `python/DCR.py`.

galactic nucleus at redshift  $z = 0$  and then at  $z = 0.5$ . We also show the transmission profiles of the LSST  $g$ ,  $r$ , and  $i$ -bands along the left side of the figure, along with the product of these two curves which represent the effective spectrum of each source as viewed through each LSST filter. Figure 2 shows the flux-weighted refraction of each SED as a function of zenith distance. This is presented in units of arcseconds; note that at the Universal Cadence limit of  $50^\circ$  zenith angle, the refraction of all sources approaches 1 arcminute. In Figure 3, we show the *differential* refraction of each spectrum, DCR, with respect to the G0V star. Note that the maximum amplitude of DCR is approximately  $0.1''$  at  $50^\circ$ , or approximately half an LSST pixel. This is the effect we wish to explore in this sequence of analyses. We included an airmass-dependence to the filter profiles using the `throughputs` package to yield the flux-weighted DCR offsets expected for each source in subtask 3, with values listed in Table 5.

### 3 Results: Subtask 1

We were unable to reproduce an *exact* match to simulated images from W13, but were able to create qualitatively similar images using `phoSim` v3.3.2 and the W13 control files, with the additional control parameter `fieldanisotropy` 0. We note that the realized seeing values of the images, as measured by the adaptive second moments of the `Psf`, are systematically ( $\sim 5\%$ ) smaller in the newer data. This was important during the analysis of the deconvolution data, as it triggered a bug resolved in Ticket #3143.



**Figure 3: Differential Chromatic Refraction vs. Filter and Spectral Energy Distribution:** The differential chromatic refraction of all sources from Figure 1 with respect to the reference G0V star, as a function of zenith distance. The maximum amplitude of DCR between the “red” and “blue” objects reaches 0.2'', or approximately one LSST pixel. This figure may be recreated using the script `python/DCR.py`.

We compare the original W13 results, the W13 data run through the W14 pipeline, and the W14 pipeline results in Table 2. We first note that the two results running the DM stack on the W13 data are not exact replicates (first vs. second column). The numbers of false positives was slightly lower in the newer analysis, especially in the better seeing data, suggesting minor but noticeable changes in the algorithms. The W14 analysis is qualitatively similar, *except* the configuration that leads to deconvolution of the template (seeing visit 1 and no prefiltering). The moderately better seeing triggered a bug in the configuration of the deconvolution kernels, which was resolved in ticket #3143 and in subtask 2. This improvement suggests both that the deconvolution configuration is currently highly sensitive to the input FWHMs, and that catastrophic deconvolution failures appear to be potentially resolvable at the configuration level (i.e. a solution is not “impossible”).

## 4 Results: Subtask 2

The numbers of false positives in the *gri*-bands are reported in Table 3. We averaged over all 9 CCDs in the raft to report the mean numbers of false positives per CCD. Figure 4 compares the number of 5-sigma detected **DiaSources** in red. The black line depicts the theoretical run of false positives with detection threshold, using the mean seeing for each sample. We find that, in general, our results are consistent with this theoretical limit. One noteworthy exception is in the deconvolution configuration, where the 0.88'' template is differenced against the 0.6'' science data without prefiltering (leftmost column of top panel). In this case, our realized false positive rate was significantly *lower* than expected from theory. This is consistent with the results from W13 (Section

Subtask 1: False Positives in `raft=2,2 sensor=1,1`

Seeing	Prefilter	W13	W14 pipeline on W13 data	W14
1	True	10	10	17
	False	23	19	68630
2	True	7	6	6
	False	8	8	6
3	True	3	3	3
	False	7	7	3

Table 2: Summary of the results of subtask 1, which aims to reproduce the results of W13 using `phoSim v3.3.2` and the current `v7_3` stack. The first column shows the numbers of false positives from `raft=2,2 sensor=1,1` of the original W13 analysis, the central column from the re-reduction of those same data using the `v7_3` stack, and the final column the results of the regenerated images during W14. The large numbers of false positives in the deconvolution configuration (seeing 1, prefilter False) were triggered by the bug resolved in ticket #3143. Results from this table were generated using script `python/countDiaSourcesT1.py` in this repository.

Subtask 2: Average False Positives in `raft=2,2`

Seeing	Filter	N	
		Prefilter=False	Prefilter=True
1	g	2	12
		7	4
		4	3
1	r	2	8
		7	5
		4	3
1	i	1	10
		5	4
		3	2

Table 3: Summary of the results of subtask 2. We show the average (over all 9 sensors of raft 2,2) of the total number of false detections per sensor at 5-sigma, both positive going and negative going. The vast majority of these sources are “orphans”; the average number that associate with a source in the image is less than 1 in all cases. Comparisons of these numbers with theoretical expectations are presented in Figure 4. Results from this table were generated using script `python/countDiaSourcesT2.py` in this repository.

6.4), where deconvolution significantly enhanced the noise properties of the difference image.

To quantify this effect in W14, we examined the variance in two ways. First, we calculated the median value of the difference image variance plane  $V_p$ , ignoring any masked pixels. We then calculated the empirical variance of the corresponding image plane, scaling its interquartile range by 0.741 as expected from a Gaussian distribution, and squared this result to yield  $V_e$ . We calculated the ratio  $V_e/V_p$  to measure how much the propagated variance over/underestimates the empirical variance. We find that the median of this ratio was within 1% of 1.0 for all prefiltering and postfiltering data, *including the deconvolution configuration*. This median value is however consistently above 1.0, indicating slight underestimation of the true variance by the propagated variance plane in all cases. This indicates that the variance plane provides an adequate estimate of the noise in the image at the stage immediately following image subtraction.

Secondly, we took the difference images that had *not* been filtered by their `Psf`, and perform the `Psf` convolution. We note that this is the image that the detection algorithm operates on in the postfiltering configuration (in fact, this is the postfiltering process itself!). In prefiltering, this detection image is the image difference product itself, which we demonstrated above to have adequate variance tracking. We then calculated the same ratios for this filtered difference image. We find that for the deconvolution configurations, the variance plane provided an *overestimate* of the empirical variance by 30–40% ( $V_e/V_p = 0.7\text{--}0.76$ ). For the convolution configurations, the variance plane *underestimates* the empirical variance by 4–5%. An overestimate of the variance by 35% should lead to 15–20 times lower false positive rate at 5-sigma; a variance underestimate by 5% should lead to a false positive rate 2–3 times higher. These numbers are roughly consistent with, but larger in amplitude than, empirical results. Figure 4 shows that the `prefilter=False` false positives rates are significantly lower than theory in the prefiltering seeing 1 data (deconvolution; the variance plane overestimates the true noise) and moderately but consistently higher than theory in the seeing 2,3 data (convolution; variance plane underestimates the true noise). As mentioned in the W13 report, the fact that the template is convolved twice is thought to be the main reason we have a poor estimate of the noise at the detection stage in the postfiltering analysis. For completeness, we found no filter-dependence to these results. We used script `python/examineVariance.py` for this analysis.

## 5 Results: Subtask 3

Figure 5 demonstrates the designed directions of increasing altitude (orientation of blue-source positive dipoles due to DCR) in visits A, B, D, and E (visit C is effectively at zenith and thus no DCR is expected). These are set by `phoSim` configuration parameter `rotTelPos`. The direction to the pole is also indicated, and set by `phoSim` configuration parameter `rotSkyPos`. For reference, `rottelpo` is defined as (`rotskypos`-180+parallactic angle), such that the angle of DCR should be along this axis. The numerical values of these angles, measured clockwise with respect to the positive y-axis (i.e. “up”) are listed in the first numerical column of Table 4. The per-spectrum and per-passbands expected amplitudes of the DCR shift, w.r.t. the GOV stars, are given in the first numerical column of Table 5.

### 5.1 Dipole Orientation and Amplitude

Our first validation of DCR comes from looking at the fitted `Wcs` of each `calexp` to establish directionality. Using a topocentric correction and the requested times of observation, we empirically determined the angles of increasing Ra,Dec and Az,Alt by taking 10” steps along each axis, starting at the field center, and converting these coordinate steps to pixel offsets. These axes are represented

Subtask 3: Orientation of DCR
-------------------------------

Visit	SED	<code>rotTelPos</code>	Wcs + Topo	Measured
				Dipole Orientation
A	AOV	145.7°	144.9°	145.7°
	M2.0V	325.7°	324.9°	325.6°
B	AOV	152.3°	151.1°	151.7°
	M2.0V	332.3°	331.1°	331.8°
D	AOV	349.9°	351.0°	349.3°
	M2.0V	169.9°	171.0°	170.0°
E	AOV	356.4°	357.1°	355.2°
	M2.0V	176.4°	177.1°	176.2°

Table 4: The expected and measured orientations of DCR, using the coordinate conventions depicted in Figure 5. These numbers represent the orientation of the *positive* lobe of any dipole arising from the DCR effect for blue sources, and reflect the angle of increasing altitude. Numbers are reported as a function of visit ABDE, and the spectral energy distribution of the source. The numbers under `rotTelPos` reflect the designed orientation of DCR that was input to `phoSim`. The numbers under Wcs reflect the empirically determined direction to zenith in each image, using the `calexp` fitted Wcs and topocentric corrections appropriate for each visit. Finally, we report the measured median orientation of DiaSources in each image difference, when differenced against the template taken at zenith (visit C). For the `rotTelPos` and Wcs columns, the numbers for the red M2.0V stars are simply 180° from those of the blue AOV stars. For the orientation column, we use fitted values from *g*-band data, seeing 1 (dipole results in the *r*-band show quantitatively similar results). We also find no dependence of these orientations on seeing. We only report these numbers using the postfiltered data, where dipole measurement is known to operate correctly. Full results may be found in the file NOTES.

in Figure 6 for visits ABDE, and the numerical values of the angle of increasing Altitude are provided in the second column of Table 4. We note that these angles are similar to the requested `rotTelPos`, typically to within a degree.

Our second validation of DCR comes from looking at the astrometric residuals of the stars with respect to the reference catalog, after fitting for the Wcs model. We grouped the stars into 3 bins based upon their SED. We then calculated the mean amplitude of residuals to the astrometric fit, which is driven by the G0V stars. The second numerical column of Table 5 provides this information. We note that the DCR amplitudes for the AOV objects are roughly consistent with theory, although uniformly over-refracted by ∼4%. The amplitudes of the relative refraction of the M2.0V objects are under-refracted but by ∼10%. The expected trend of the “red” star DCR amplitude being larger than the “blue” stars amplitude, as depicted in Figure 3, is seen here.

Finally, we compared the measurements of dipoles in the difference images to the expected effects of DCR. For this analysis, we only looked at difference images where the template image comes from zenith (i.e. visit C). The joint-Psf dipole measurement model simultaneously fits for a positive-going and negative-going component of the source – a 6-parameter fit including 2 centroids and 2 fluxes. The separation of the positive-going and negative-going component are proxies for the DCR effect. We used measurements from the postfiltered data only because the measurement suite is not fully implemented for prefiltered data. Because we found the most dipoles in the *g*-band seeing 1 data, we used these measurements to validate the DCR orientation. The orientation

Subtask 3: Amplitude of DCR

Visit	SED	Filter	Theory	Wcs Offsets	Dipole Fit Seeing Bin 1	Dipole Fit Seeing Bin 2	Dipole Fit Seeing Bin 3
A	AOV	<i>g</i>	0.0406"	0.0421±0.0014"	0.167"	0.259"	0.208"
		<i>r</i>	0.0125"	0.0129±0.0005"	0.049"	...	...
		<i>i</i>	0.0040"	0.0044±0.0005"	...	...	...
	M2.0V	<i>g</i>	0.0862"	0.0768±0.0029"	0.144"	0.204"	0.293"
		<i>r</i>	0.0207"	0.0199±0.0007"	0.230"	...	...
		<i>i</i>	0.0125"	0.0120±0.0008"	...	...	...
B	AOV	<i>g</i>	0.0199"	0.0205±0.0008"	0.183"	0.243"	...
		<i>r</i>	0.0060"	0.0064±0.0007"	...	...	...
		<i>i</i>	0.0019"	0.0023±0.0004"	...	...	...
	M2.0V	<i>g</i>	0.0424"	0.0385±0.0012"	0.222"	0.266"	...
		<i>r</i>	0.0101"	0.0099±0.0007"	...	...	...
		<i>i</i>	0.0061"	0.0056±0.0008"	...	...	...
D	AOV	<i>g</i>	0.0199"	0.0200±0.0008"	0.201"	0.251"	...
		<i>r</i>	0.0060"	0.0062±0.0006"	...	...	...
		<i>i</i>	0.0019"	0.0025±0.0005"	...	...	...
	M2.0V	<i>g</i>	0.0424"	0.0385±0.0010"	0.201"	0.254"	...
		<i>r</i>	0.0101"	0.0097±0.0006"	...	...	...
		<i>i</i>	0.0061"	0.0057±0.0008"	...	...	...
E	AOV	<i>g</i>	0.0406"	0.0430±0.0027"	0.169"	0.235"	0.360"
		<i>r</i>	0.0125"	0.0127±0.0004"	0.135"	...	...
		<i>i</i>	0.0040"	0.0049±0.0007"	...	...	...
	M2.0V	<i>g</i>	0.0862"	0.0771±0.0022"	0.158"	0.213"	0.267"
		<i>r</i>	0.0207"	0.0197±0.0006"	0.262"	...	...
		<i>i</i>	0.0125"	0.0115±0.0010"	...	...	...

Table 5: The expected and measured amplitudes of DCR, in arcseconds. These numbers represent the differential offset between positions of an object at zenith (visit C) and at the airmasses associated with visits ABDE, with respect to the positions of the reference GOV stars, which define the astrometric reference system in this study. The Theory column is the expected amplitude as described in Section 2, using airmass-dependent filter profiles, atmospheric water vapor pressure of  $f = 8$  mmHg, ambient air pressure of  $P = 520$  mmHg, and ground temperature of  $T = 20$  C. The Wcs column represents the median (over all seeing values and sensors) offsets between stars of the given SED and the astrometric reference solution, determined using the Wcs and Source products. Results in all passbands are consistent with zero refraction for the CC visits (within 0.001"). We find moderate airmass dependence of the GOV star residuals with an amplitude of 0.002", 0.0005", and 0.0006" in the *g, r, i*-bands, respectively. The mean dipole amplitudes are determined using the same fits that yield the dipole orientations in Table 4, and represent the offset between the fitted positive and negative lobes of the dipole. Because we see a seeing dependence on these numbers, we report them for each of the seeing bins 1, 2, 3. Theory results were determined using script `python/DCR2.py`. Astrometric residuals were determined using script `astrometricResids.py`. Full results may be found in the file NOTES.

numbers are presented in the third numerical column of Table 4. We find that these numbers are also consistent with the designed orientation to within approximately  $1^\circ$ . We find no seeing dependence on these values.

The passband-dependent separations of the dipole lobes are given in the final columns of Table 5, again only using the data differenced against template visit C. We find that these values are considerably larger than the DCR amplitude expected from theory, and from the amplitudes suggested by the `Wcs` residuals. Additionally, we find that this overestimate is a function of seeing, in the sense that in the worse seeing data these values are typically overmeasured by a larger degree. This is a strong indication that there are aspects of the dipole measurement algorithm that need to be improved upon. We only found significant numbers of dipoles in the  $r$ -band data in the data at airmass 1.55. For the  $i$ -band data, there were not enough dipoles in any conditions to provide a measurement.

## 5.2 Dipole Rate

Figures 7 and 8 show the median number of false positives per CCD within raft 2,2. These are given as a function of passband, and science visit vs. template visit. The “on-axis” elements within each matrix represent configurations where DCR within the template and science image are exactly aligned. These values are consistent with the results seen in subtasks 1 and 2, indicating that DCR is an effect that may be compensated for when realized similarly in a template and science image.

The off axis elements correspond to a mismatch between template and science image conditions, and show an enhanced rate of false positives. This enhancement is strongest in the  $g$ -band, and in the better seeing data. The number of false positives is also significantly smaller in the prefiltering data. Importantly, when the airmasses are similar but the parallactic angle through the images different, an enhanced rate of false positives is seen, suggesting that the image subtraction template suite must accommodate orientation as well as airmass. We elaborate on this in Section 6. The effects of mismatched parallactic angles are mostly negligible in the  $i$ -band data, except in the best seeing conditions. However, for the majority of the  $g$  and  $r$ -band (and presumably  $u$ -band) data, DCR appears to be a significant impediment to achieving a noise-limited false positive rate.

There are 2 apparent components to the heat maps. The first corresponds to an enhancement in the rate of the no-refraction to high-refraction differences (CA and CE), which show large numbers of false positives in good seeing, and in the prefiltering data appears localized to the  $g$ -band data. The second component corresponds to the maximal mismatch between the DCR vector (AE and EA), and appears to be the dominant mode in the worse seeing data, in the redder passbands, and in the non- $g$ -band prefiltering data.

## 6 Implications for Image Subtraction Templates

We examine the implications for the image subtraction template suite, using the observed correlation between DCR amplitude and false positive rate.

We first look at the number of false positives when differencing against the zenith template C. We associated these false positives with the reference catalog, and binned them based upon the SED they match to. We then normalize each of these populations by the number that are expected from background fluctuations. We plot these values against the measured DCR offsets (using the WCS-based analysis from Table 5), in units of the PSF FWHM, in Figure 9. Data from the AOV stars are plotted in blue, and from the M2.OV stars in red. We use information from all passbands and all seeings on this plot; the seeing 1,2,3 data are encoded as circles, squares, and triangles, respectively.

We fit a linear regression to each curve of the form  $\log_{10} y = a + b * \log_{10} x$ , finding:

$$\begin{aligned}
\text{AOV Prefilter = True } & \log_{10} y = 6.0913 + 2.2470 \log_{10} x \\
\text{Prefilter = False } & \log_{10} y = 7.4561 + 2.7537 \log_{10} x \\
\text{M2.0V Prefilter = True } & \log_{10} y = 6.4141 + 2.6279 \log_{10} x \\
\text{Prefilter = False } & \log_{10} y = 6.5785 + 2.6713 \log_{10} x
\end{aligned} \tag{6}$$

We plot these as the solid lines in Figure 9. The curves are significantly different for the red and blue stars, which is not entirely understood at this time. We naively expect the curves to look the same since we are effectively including information about the spectral type in the amplitude of the DCR shift, along the x-axis. It is possible that these difference come from second-order color effects in the sims, including color-dependent shape and size of the PSF. Vertical lines are plotted where these curves cross  $y=1$ , which effectively represents a doubling of the false positive rate due to DCR from each class of stars. We find that the prefiltering and postfiltering lines cross at essentially the same value, amplitude/FWHM = 0.0019 and 0.0034 for the AOV and M2.0V stars, respectively. For a fiducial seeing value of 0.6'', this means that DCR begins to dominate the false positive rate at around DCR amplitudes of 1–2 mas.

We next examined the consequences of this on the airmass and parallactic angle requirements for image subtraction templates. We first calculated the DCR for each source star at a fiducial airmass of 1.25, corresponding to the first leg of a triangle. The second leg is calculated by considering the amplitude *and* orientation of DCR at a range of airmasses from 1..1.5, and orientation differences from 1..180 degrees. The airmass is used to calculate the length of the second leg of the triangle; we assume that the intersection of these 2 legs represents a coordinate system defined here by the GOV stars, such that these vectors represent 2 different realizations of DCR. The opening angle between these two vectors is set to 0..180 degrees in steps of 10, and represents the mismatch between the parallactic angle in the 2 images. We use the law of cosines to describe the astrometric offset between the two apparitions of the source (at airmass 1.25, and at all other combinations of airmass and parallactic angle). This offset is used to color-code the panels in Figure 10, which show DCR amplitude as a function of airmass along the y-axis, and difference in the parallactic angle along the x-axis. We overplot lines of constant offset at 1mas and 2mas, which we found in Figure 9 corresponds roughly to the DCR amplitudes that cause significant false positive rates for the AOV and M2.0V stars, respectively. Formally, we should only consider the blue lines in the AOV plots, and the red lines in the M2.0V plots. However, since we don't fully understand the reasons for these curves being different, we plot both lines in all figures. We note that Figure 10 suggests that, for the reddest objects in the *g*-band, a difference in parallactic angle of 20–30 degrees is sufficient to start driving the false positive rate with DCR dipoles. For the bluest objects, a similar effect is seen starting at parallactic angle differences of 40–50 degrees. For the *r*-band, and only for the reddest objects, a difference in parallactic angle of 90 degrees is required before DCR drives the false positive rate. We emphasize that these values are w.r.t. a template at 1.25 airmasses; we see slight differences in these curves when comparing to reference observations at 1.0 and 1.5 airmasses due to the non-linearity of the refraction vs. airmass curve.

## 7 Thoughts on the DCR Issue

We outline possible resolutions to the DCR issue. Classes that need to know about DCR include the `WCS` class, for astrometric modeling, and the `Psf` class, for its contribution to the color-dependent

point spread function. The consumers of this information include the image coaddition and differencing tasks. The latter has a double dependency on DCR, since it uses coadded templates whose generation themselves depends on an adequate treatment of DCR. We expect to work towards this resolution in the upcoming Summer 2014 production (S14) cycle.

Modeling the expected effects of DCR should be relatively straightforward, assuming the colors of most objects are known from data release processing, the observation conditions are known exactly, and the environmental conditions known approximately. The majority of objects whose instantaneous color is not known (moving objects, or objects that change in brightness) will leave signatures in difference images anyways, so not knowing their color should not create additional “false positives”. It may however complicate the measurement of these objects. Places where this assumption breaks down completely are for objects that are different in color but not brightness, or for crowded fields where every pixel has its own color.

Compensating for the effects of DCR within an image requires not the shifting of the image, but the shifting of every object *within* the image along the vector to zenith. This suggests a color-dependence to the pixel resampling process, which might use both the `Wcs` and a putative `Dcr` class to map the location of an object to unrefracted coordinates (i.e. above the atmosphere). This would be extremely important if we wished to “undo” the effects of DCR before coadding images taken at different airmasses and parallactic angles. We would also have to “mimic” DCR when mapping coadded image subtraction templates to the science image conditions (assuming that all pixel-level operations happen on the template during image subtraction). In these scenarios, neighboring pixels must be treated as a bundle, to preserve the `Psf`. It is unclear how to proceed when remapping e.g. a nearby galaxy, or a crowded starfield in the Galactic plane.

The simplest solution is to reject objects (or pixels) of extreme color from triggering alerts at high airmass. A slightly more complicated solution is to include the effects of DCR in measurement, i.e. improve the dipole measurement code. And the most complicated option is the one described above, where DCR is compensated for at the per-object pixel level, either during pixel remapping/resampling or during `Psf/Psf`-matching kernel modeling. We can take some guidance from the analysis presented in Section 3.3 of Alcock et al. (1999), who use a per-pixel color map to compensate for DCR. A necessary improvement over that technique is the need to undertake this operation in a way that preserves the PSF. A final option is to make this a consideration during the acquisition of the data (e.g. observing a field within a narrow range of azimuth), the feasibility of which may be tested via OpSim.

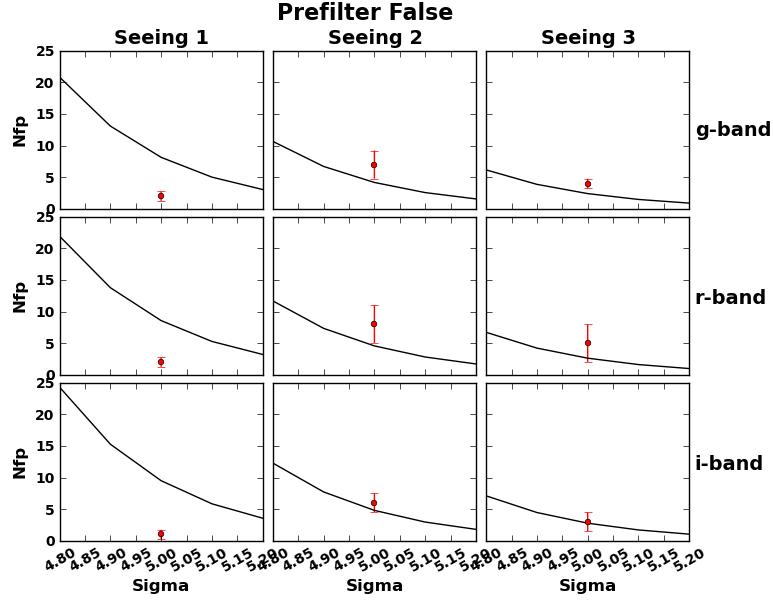
Other options are both solicited and welcome.

## 8 Conclusions

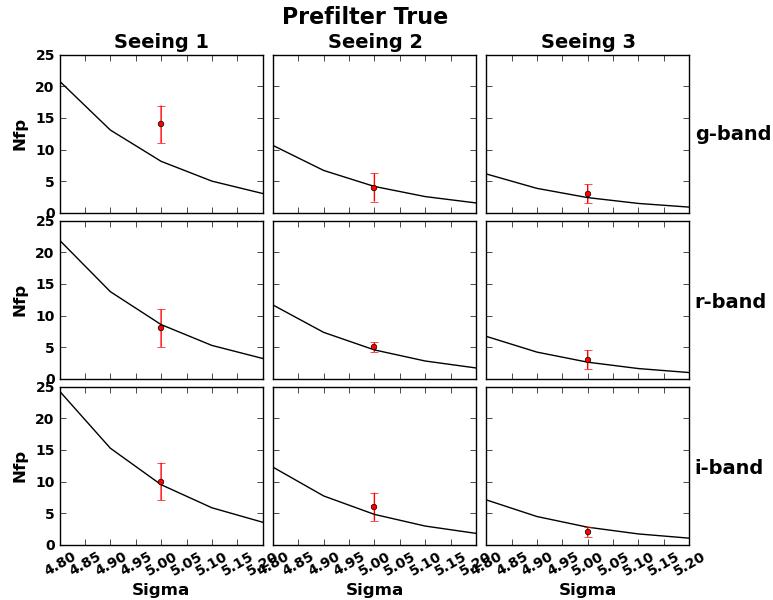
- We are able to qualitatively reproduce the results of W13, namely a false positive rate in difference images that is near the theoretical rate expected from fluctuations in the background. Quantitative differences reflect changes in both `phoSim` and the DM stack since W13. We find no filter-dependence to these results.
- We find that the variance planes of the data products immediately following image subtraction are consistent with the empirical variance to 1%. This result holds for both the prefiltering and postfiltering analysis paths.
- However, the convolution of an unfiltered difference image with its `Psf` (i.e. the postfiltering process immediately preceding detection) results in an image whose variance plane provides a poor estimate of the true variance. In the case of deconvolution of the template, this variance

plane *overestimates* the true variance by 30–40%, resulting in a lower rate of false positives but also shallower effective depth. In the case of smoothing of the template, this variance is a 4–5% *underestimate*, resulting in an enhanced rate of false positives. This effect is only relevant for the postfiltering analysis path.

- We find that differencing templates and images taken at the same airmass and parallactic angle – regardless of airmass, parallactic angle, filter, or seeing – generally produce the same high quality results.
- We find that mismatches between parallactic angle, even for images taken at the same airmass, are sufficient to produce significantly enhanced rates of false positives in difference images, especially in the  $g$ –band and  $r$ –band. Mismatches between both airmass and parallactic angle also yield enhanced false positive rates. Accordingly, DCR is a substantial issue that needs to be addressed.
- This enhanced false positive rate is more significant in the bluer passbands, but declines significantly by  $i$ –band.
- The enhanced rate is also seeing dependent, in the sense that better seeing data yield more false positives than worse seeing data.
- Two components in the false positive “heat maps” correspond to enhanced false positive rates in the no–fraction vs high–refraction data (CA and CE differences) and in the data with maximal mismatch between the DCR vectors (**AE** and **EA**).
- While the enhanced false positive rate is lower in the prefiltered difference images than in the postfiltered, it is not negligible.
- DCR of 1–2 milliarcseconds appears sufficient to produce a false positive rate similar to that expected from Gaussian background fluctuations, at a reference seeing of 0.6”.
- This translates into a requirement that  $g$ –band templates not have differences in parallactic angle of more than 20–30 degrees, to avoid substantial DCR from the reddest stars (assuming a coordinate system defined by G0V stars). This requirement is loosened to one every  $\sim 90$  degrees in the  $r$ –band.
- The orientation of the DCR effect is consistent with expectations.
- We find that the amplitude of DCR shift in the red M2.0V stars is under-realized by  $\sim 10\%$  compared to expectations. A0V stars are over-refracted by  $\sim 4\%$  compared to expectations.
- Dipole measurement measures lobe separations that are substantially larger than the effects of DCR. This effect gets worse with seeing, indicating a potential need to improve the algorithm.
- The treatment of DCR will likely be a complicated one, and at the extreme may require an object–dependent remapping during the image registration process. Image subtraction is doubly dependent on this effect, as its template images need to first be coadded (which requires registration of input images *and* of the DCR–affected objects within the images) before being warped to the image coordinate system to be used for subtraction.
- We further document the end–to–end process of creating and running simulated data in the Appendices below.

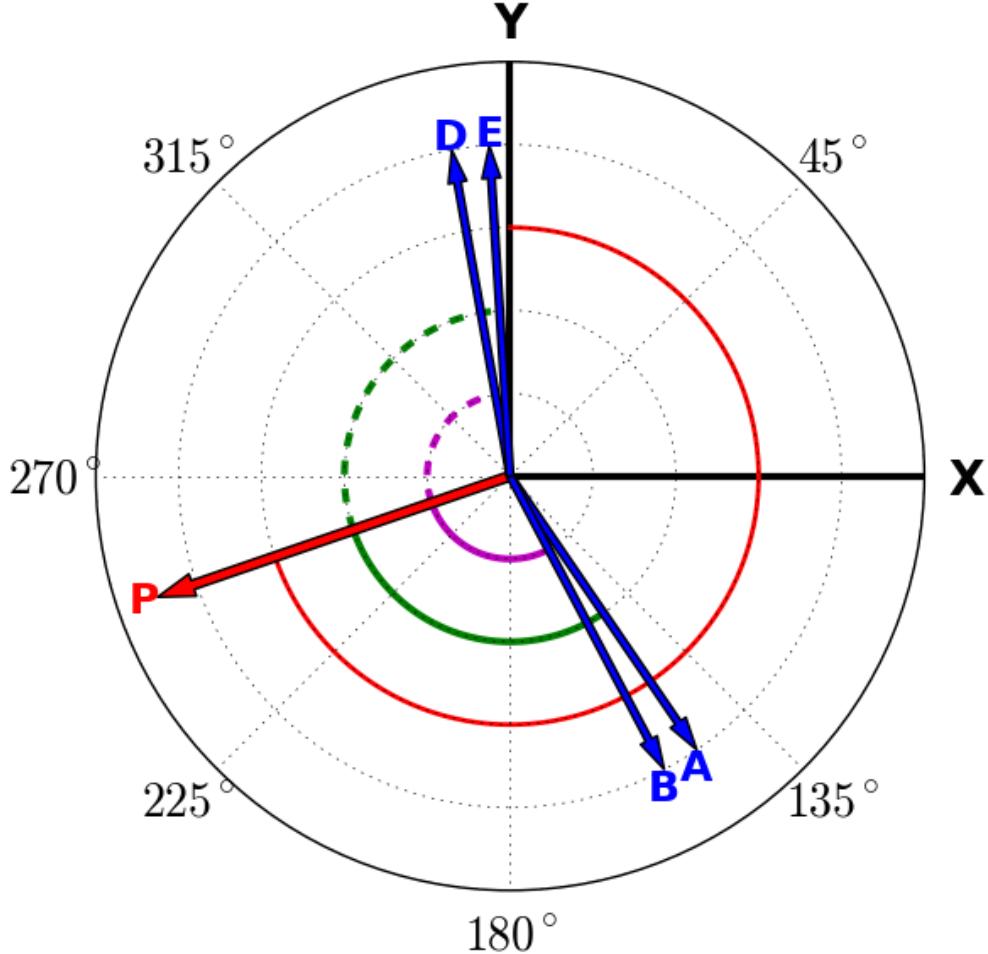


(a) Prefiltering = False

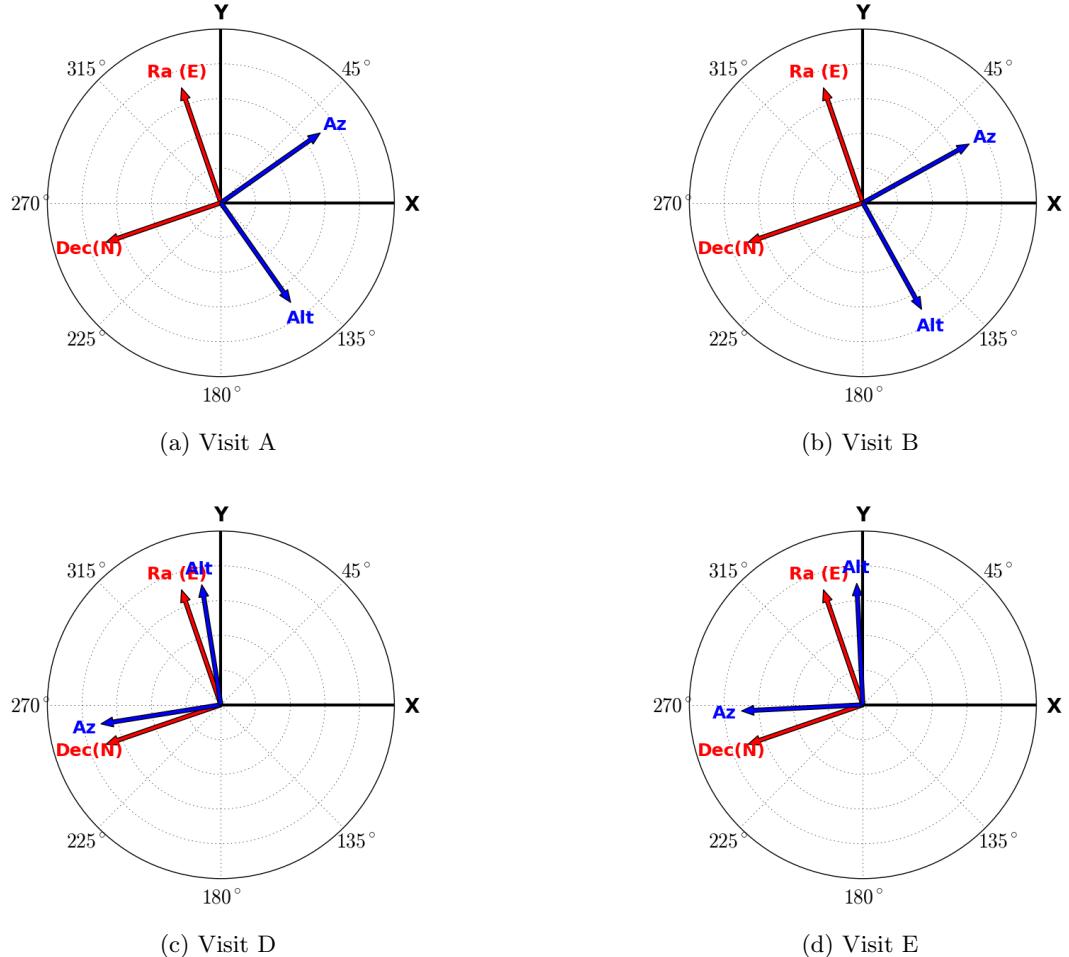


(b) Prefiltering = True

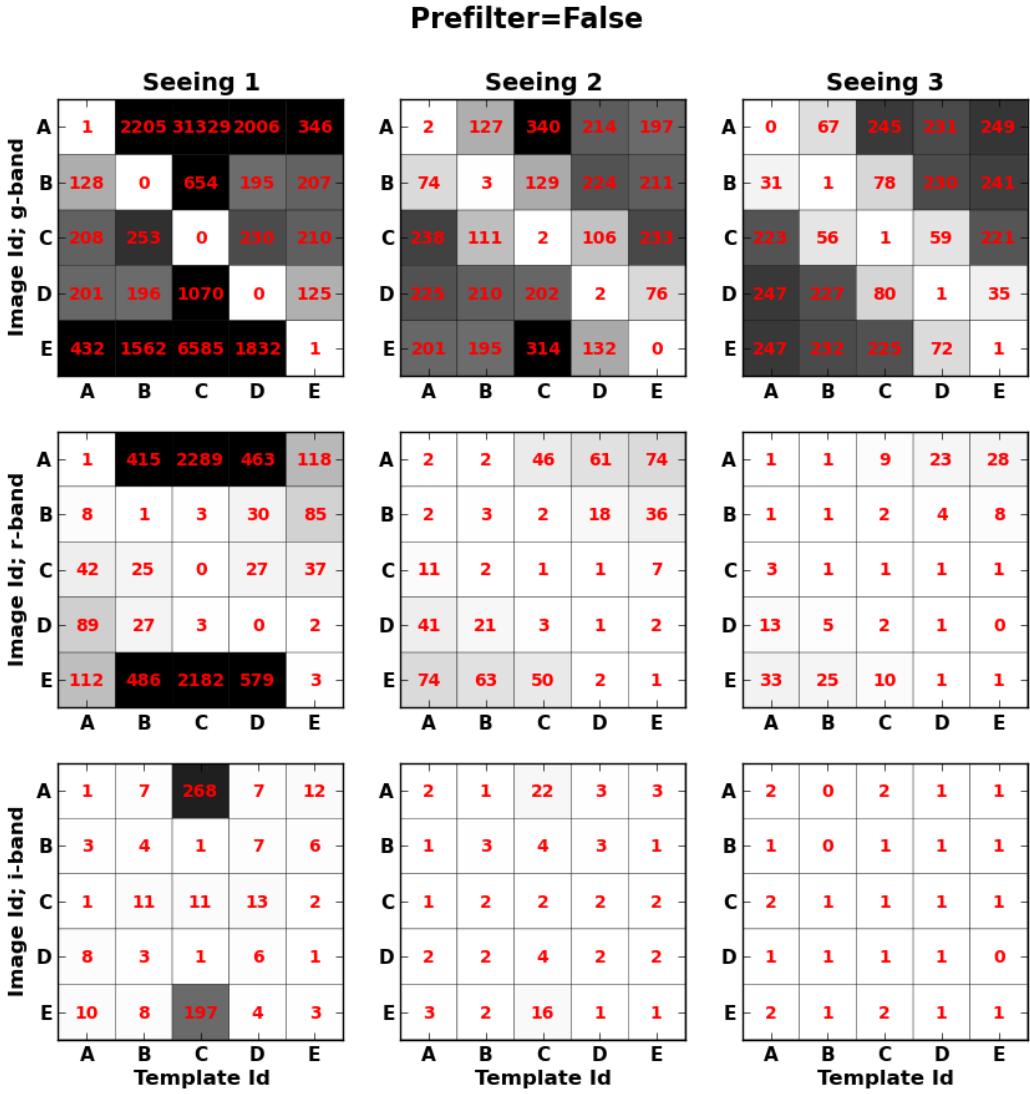
**Figure 4: Subtask 2: False Positive Rates vs. Theory:** Using the mean realized seeing for each passband and `Opsim_rawseeing` combination, we plot the expected numbers of false positives per sensor (both positive and negative going) as a function of detection threshold. We plot the median number of DiaSources detected at 5 sigma across the nine `raft=2,2` CCDs in red, using the interquartile range to calculate the RMS. Results using postfiltering of the data with its `Psf` are on the top, and prefiltering are on the bottom. Note that the postfiltering false positives in seeing 1 which correspond to a template deconvolution, are lower than expected due to an overestimate of the noise by the variance plane. The postfiltering positives for seeing 2,3 are consistently above the theoretical curve, likely due to a 4–5% underestimate of the noise by the variance plane. Prefiltering false positives appear more evenly distributed above and below the theoretical curves. This figure was created using the script `python/falsePositivePlots.py`.



**Figure 5: Designed Orientation of DCR in W14 PhoSim Runs:** This figure represents the anticipated orientations of DCR in the W14 phoSim data. The x,y coordinate system is depicted, as well as the convention that angles (`rotTelPos`, `rotSkyPos`) rotate clockwise with respect to the positive y-axis in the image coordinate system (and counterclockwise in the camera coordinate system). The `rotTelPos` of 251 degrees specified for all simulations, which reflects the direction to the pole, is shown with the red vector P and the red arc at  $y=0.6$ . The derived `rotSkyPos` for visits A,B,D,E are shown with the blue vectors, and reflect the angle towards zenith (the angle of increasing altitude). DCR is expected to happen along these vectors. The angles PA,PE are similar, and represented by the green arcs at  $y=0.4$ ; the angles PB,PD are also similar, and represented by the purple arcs at  $y=0.2$ . This is expected as observations A and E are taken at airmass 1.55 (zenith distance of 50 degrees) but at opposite sides of the meridian crossing of the star field; a similar situation was designed for observations B and D, which are taken at airmass 1.16 (zenith distance of 30 degrees). Visit C is not depicted as it was taken at zenith. This figure was created using the script `python/dcrSchematic.py`.



**Figure 6: Wcs-Derived Orientations of PhoSim Data:** These figures show the orientations of the Right Ascension and Declination axes (red), and Azimuth and Altitude axes (blue) of visits A,B,D,E. Arrows represent the directions of *increasing* coordinate value. The Ra,Dec axes are the same in all images since they were designed to have a common `rotTelPos`. Ideally, the directions of increasing Alt will correspond to the `rotSkyPos` depicted in Figure 5. All coordinate system orientations were derived from the fitted Wcs of the `calexp` of the *g*-band observation of seeing 2 data. To determine the orientations empirically, small steps were taken in each coordinate starting at the center of the image, and the Wcs and topocentric corrections used to map these back into offsets in the pixel plane. This figure was created using the script `python/compareDcrFromSims.py`.



**Figure 7: Number of False Positives, Postfiltering:** This figure shows the median number of false positives across the nine `raft=2,2` CCDs. The first row of information shows these “heatmaps” for *g*–band data, the second for *r*–band, and the third for *i*–band. The first column represents the good–seeing images, the second the medium–seeing images (same quality as template), and the third the poor–seeing images. Within each filter–seeing combination, the heat–map represents the median number of false positives as a function of the image airmass (visit ABCDE) along the y–axis, and template airmass (visit ABCDE) along the x–axis. The diagonal elements represent the situation where the template and science image are taken at the same airmass and have the same orientation w.r.t. zenith. The off–diagonal elements represent a mismatch between the template and science image in terms of airmass and/or parallactic angle. The general trend is that the numbers of false positives decrease with decreasing differences in the airmass,angle attributes of the template and science image, decrease with increasing seeing, and decrease with increasing wavelength. The *i*–band data in poor seeing do not appear sensitive to DCR. This figure was created using the script `python/heatMap.py`.

### Prefilter=True

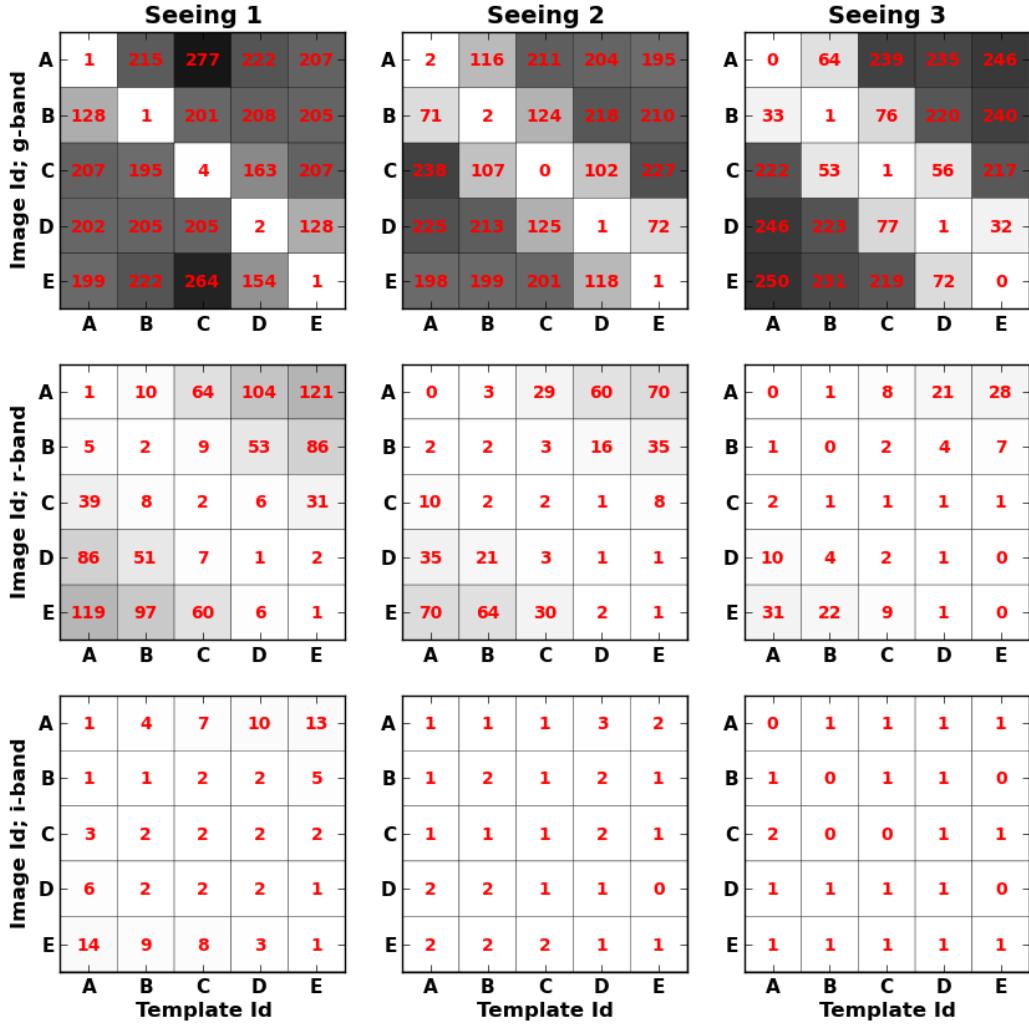
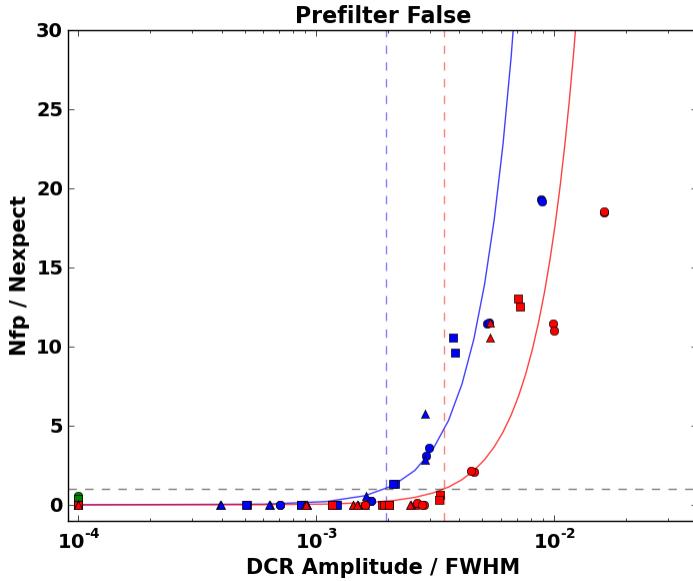
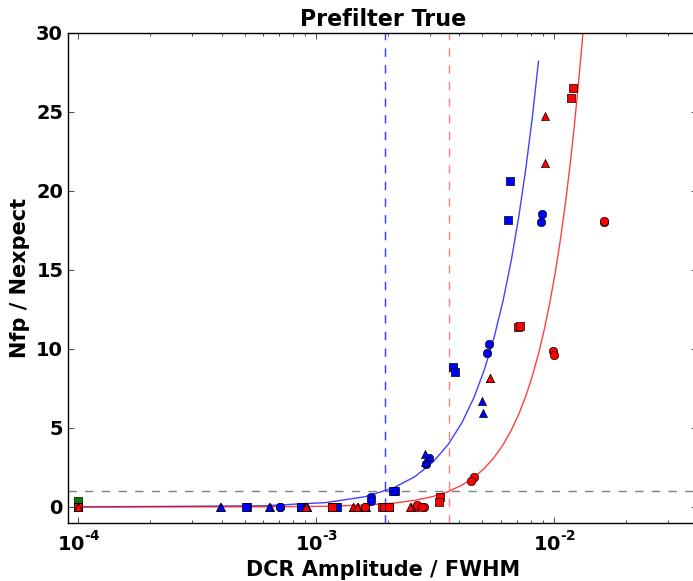


Figure 8: **Number of False Positives, Prefiltering:** Same as Figure 7, but using prefiltering of the science image with its Psf. The numbers of false positives is generally lower than in the postfiltering case (Figure 7). This figure was created using the script `python/heatMap.py`.

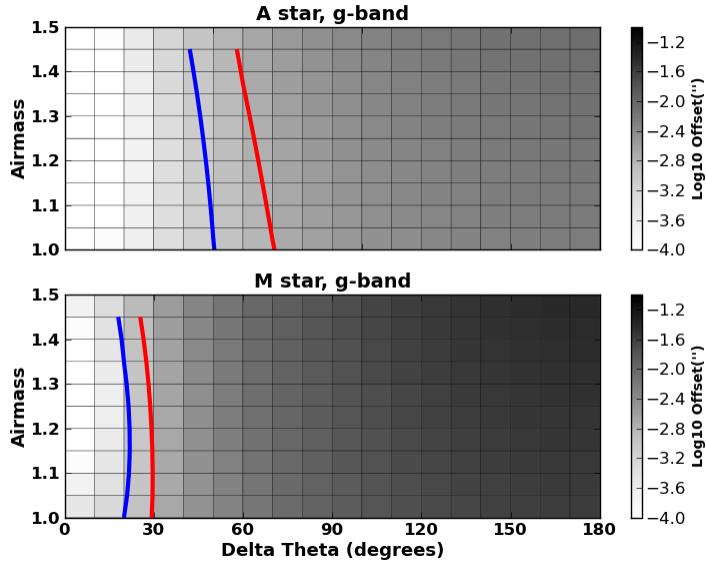


(a) Prefiltering = False

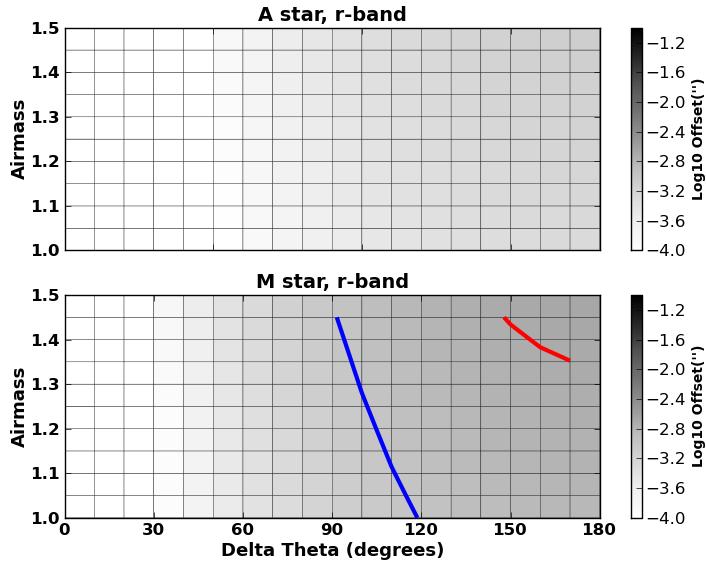


(b) Prefiltering = True

**Figure 9: DCR Impact on False Positives:** We plot the number of false positives that associate with AOV stars as the blue points, and M2.0V stars as the red points. We normalize by the number expected due to fluctuations in the background, such that  $y=1$  (horizontal black line) represents a doubling of false positives. We plot this value as a function of the measured DCR offset determined via WCS residuals (Table 5), normalized by the FWHM of the PSF. We collapse information from all 3 passbands on this figure; the  $i$ -band data tend to have the smaller  $x$ -values, and the  $g$ -band data larger values. We additionally include information from all 3 seeings by plotting seeing 1 data as *circles*, seeing 2 as *squares*, and seeing 3 as *triangles*. We plot regressions of the form  $\log_{10} y = a + b * \log_{10} x$  as solid lines, and the positions where these curves cross  $y=1$  as vertical dashed lines. This figure was created using the script `python/fpRegressions.py`.



(a)  $g$ -band



(b)  $r$ -band

Figure 10: **DCR Offsets vs. Airmass and Orientation:** Each of these panels shows the amplitude of the DCR vector in arcseconds, calculated as a function of passband ( $g$ -band top panel,  $r$ -band bottom panel) and spectral type (AOV star top subpanel in each, M2.0V star bottom subpanel). We start with a reference observation of each star at airmass of 1.25, and calculate the amplitude of the relative DCR vector assuming observations at other airmasses (encoded by the ordinate value) and differences in the parallactic angle orientation (abscissa). We assume the astrometric reference frame does not itself shift due to DCR. Contours of constant offsets are plotted at 1 mas in blue, and 2 mas in red, and approximately correspond to the offsets where DCR starts to dominate the the false positive rate (Figure 9) at a fiducial seeing of  $0.6''$ , the blue line from the AOV-star analysis and red from the M2.0V. This figure was created using the script `python/DCR3.py`.

## References

- Alcock, C., et al. 1999, *ApJ*, 521, 602  
Filippenko, A. V. 1982, *PASP*, 94, 715

# Appendices

The root directory for this analysis is located on the NCSA machines at `/nfs/lsst/home/becker/Winter2014`. All files not included in this `git` repository may be found there, including the output repositories needed by the majority of the analysis and plotting scripts. A complete guide to every step in this W14 process may be found in the `NOTES` file.

## A Ticket Work

### A.1 Ticket #3128

This ticket was opened to allow the user to select the sample of “red” and “blue” stars to be added to the control sample. This enabled after-the-fact diagnostics of the different populations of objects.

### A.2 Ticket #3143

This ticket was opened to fix a bug in the settings of the deconvolution kernel sizes. A variable was unintentionally causing a misconfiguration of the kernel basis in the  $g$  and  $r$ -band postfiltering seeing 1 data (which result in a deconvolution). This resulted in a factor of 10–100 increase in the deconvolution false positives, compared to the  $i$ -band. This misconfiguration results in the large numbers of false positives reported in Section 3 below, and was fixed for the analysis of Section 4. This indicates that deconvolution may be undertaken with the proper kernel configuration, resulting in an acceptable false positive rate, although the variance in the images ends up being much higher (and the effective detection threshold much lower) compared to the prefiltering route.

### A.3 Ticket #3161

This ticket was the most extensive implemented in this W14 work. Previously, joint-Psf dipole measurement (simultaneously fitting for a positive and negative-going Psf with 6 model terms including two x,y centroids and two fluxes) used a very coarse “minimization” routine. This routine effectively took half pixel steps in the 4 centroid positions, and performed a linear fit for the two fluxes at each step, returning the step that yielded the lowest  $\chi^2$ . However, the dipole separations expected from DCR were in general far smaller than the default step size, and thus the measurements were too coarse for this task. In addition, this process was slow and inefficient, making measurement of more than  $\sim 200$  dipoles in a given image infeasible.

In Ticket #3161 we implemented a fully non-linear joint Psf fit, using the `Minuit` package. This substantially sped up the per-dipole measurement process by a factor of 20–30, from 1–2 measurements per second to 20–60 per second. This speed-up analysis was performed using script `python/compareMeasurementTiming.py` in this repository. Dipole measurement is currently *not* the tall pole in processing; resources should be spent on speeding up the convolution and warping code to minimize the overall run time.

## B Use of PhoSim

I describe here the end-to-end process of generating instance catalogs (or “trim files”) for `phoSim`, running these images to create output `eImages`, creating `astrometry.net` index files for astrometric calibration, and running these images through `processEimage.py`. This does *not* include the

process of generating instance catalogs from a master base catalog, or the process of using calibration data to run the simulated images through `processCcd.py`, which does (amongst other operations) the assembly of amp images into CCD images and instrument signature removal.

The process is valid for `phoSim` version 3.3.2.

## B.1 Setting up PhoSim

To download, build, and setup `phoSim`:

```
wget https://dev.lsstcorp.org/cgit/LSST/sims/phosim.git/snapshot/phosim-3.3.2.tar.gz
tar -xzvf phosim-3.3.2.tar.gz
cd phosim-3.3.2
./configure
n
c
$CFITSIO_DIR/lib/
$CFITSIO_DIR/include/
$FFTW_DIR/lib/
$FFTW_DIR/include
make
mkdir ups
echo 'envPrepend (PATH, ${PRODUCT_DIR})' >> ups/phosim.table
echo 'envPrepend (PATH, ${PRODUCT_DIR}/bin)' >> ups/phosim.table
setup -k -r .
```

Note that you will have to declare the locations of `cfitsio` and `fftw3` by hand, and must enter the full paths above (i.e. not the string “`$CFITSIO_DIR/lib/`” but the path that it resolves to). The above assumes that you have a DM stack already, with `cfitsio` and `fftw3` setup.

## B.2 Running `phosim.py`

To run `phoSim`, you need the environment variable `CAT_SHARE_DATA` declared, as this is where the package looks for its SEDs. Example locations are, at NCSA, `/lsst/home/aconnolly/imsim/starSED/`. You also need to create `work/` and `out/` directories for the script to use; these must be made *before* the simulation is run. A “trim” file or instance catalog must be created, which contains a random seed and observation information such as pointing, seeing, and the star catalog itself. These are included in this repository for all sims runs. A control file (named `clean.params.beta` here) must also be included, and turns off/on various effects in the sims, such as sky backgrounds and tracking errors. A call to the `phoSim` package then looks like:

```
python $PHOSIM_DIR/phosim.py $PWD/2001000.trim -w $PWD/2001000.work -o $PWD/2001000.out
-c $PWD/clean.params.beta
-s "R22_S00|R22_S01|R22_S02|R22_S10|R22_S11|R22_S12|R22_S20|R22_S21|R22_S22" -p 16
```

where `-w` defines the work directory (which must already exist), `-o` the output directory (which must already exist), `-c` the control file, `-s` the sensors to simulate, and `-p` the number of cores to use.

Once the raw amplifier and `eImages` are created, they need to be renamed for consistency with the `obs_lsstSim` mapper. The repository script `python/rename_images.py` is used for this, with an example command that moves the `eImages` from the `phoSim` output directory to a DM input directory called `inputs8`:

```

cd sims8/
foreach i (*.out)
  cd $i
  python ../../python/rename_images.py ../../inputs8 *E000.fits.gz
  cd ../
end

```

Next, an input registry must be generated, and a mapper file created:

```

cd inputs8/
python $OBS_LSSTSIM_DIR/bin/genInputRegistry.py .
echo "lsst.obs.lsstSim.lsstSimMapper" >> _mapper

```

### B.3 Creating Astrometry Index Files

A calibration file must be created for `astrometry.net`, which is used as the photometric and astrometric reference. Typically this is created from a base catalog, but since I am starting with instance catalogs here, the process is somewhat different. I use one of the input trim files for this process as follows:

```

mkdir astrometry_net_data8
python python/trim_to_ref.py sims8/2002000.trim > astrometry_net_data8/2002000.ref
cd astrometry_net_data8
setup -r /lsst/home/krughoff/lsst/lsst_devel/Linux64/astrometry.net-0.38/astrometry
text2fits.py 2002000.ref 2002000.fits
setenv P 1402120
build-index -i 2002000.fits -o index-${P}00.fits -I ${P}00 -P 0 -S r -n 100 -L 20 -E -j 0.4 -r 1
build-index -1 index-${P}00.fits -o index-${P}01.fits -I ${P}01 -P 1 -S r -L 20 -E -M -j 0.4
build-index -1 index-${P}00.fits -o index-${P}02.fits -I ${P}02 -P 2 -S r -L 20 -E -M -j 0.4
build-index -1 index-${P}00.fits -o index-${P}03.fits -I ${P}03 -P 3 -S r -L 20 -E -M -j 0.4
build-index -1 index-${P}00.fits -o index-${P}04.fits -I ${P}04 -P 4 -S r -L 20 -E -M -j 0.4
ls index-1402120* | awk '{printf("modhead %s+7 REFCAT '2002000'\n", $1)}' | sh
setup astrometry_net 0.30
cp -r /nfs/lsst/home/becker/Winter2014/astrometry_net_data/ups .
setup -k -r .

```

Note that the file `andConfig.py` must be modified to use the naming convention defined by environment variable `P` above:

```

cat astrometry_net_data8/andConfig.py
root.starGalaxyColumn = "starnotgal"
filters = ('u', 'g', 'r', 'i', 'z', 'y')
root.magColumnMap = dict([(f,f) for f in filters])
root.magErrorColumnMap = dict([(f, f + '_err') for f in filters])
root.indexFiles = [
    'index-140212000.fits',
    'index-140212001.fits',
    'index-140212002.fits',
    'index-140212003.fits',
    'index-140212004.fits',
]

```

## B.4 Running processEimage.py

Finally, the running of SFM via `processEimage.py`:

```
$OBS_LSSTSIM_DIR/bin/processEimage.py . --id visit=1000000^1001000^ .... -j 20
```

where `-j 20` provides the number of concurrent threads.

## B.5 Running imageDifferenceWinter2013.py

To difference all 3 seeing images from visit A with a template from visit C, use the following command:

```
$PIPE_TASKS_DIR/bin/imageDifferenceWinter2013.py inputs8 --id
visit=3000001^3000002^3000003
--output=/nfs/lsst/home/becker/Winter2014/outputs8CA_doPreConvolveFalse
--configfile imageDifferenceConfigGstar.py --config diaSourceMatchRadius=2.0
doAddCalexpBackground=False winter2013TemplateId=3002000 doPreConvolve=False
subtract.kernel.active.singleKernelClipping=False
subtract.kernel.active.kernelSumClipping=False
subtract.kernel.active.spatialKernelClipping=False
subtract.kernel.active.scaleByFwhm=True
subtract.kernel.active.spatialKernelOrder=5 subtract.kernel.active.alardNGauss=3
subtract.kernel.active.alardDegGauss="[5,2,2]"
subtract.kernel.active.alardSigGauss="[1.0,1.0,1.0]"
doSelectDcrCatalog=True --clobber-config -j 27
```

Important flags here are the W14 configuration file `imageDifferenceConfigGstar.py`, the template visit via `winter2013TemplateId`, and the prefiltering flag via `doPreConvolve`.

## C Generation of Instance Catalogs for W14

We outline the salient (and non-obvious) configuration parameters for the W14 trim/instance catalogs below:

- `SIM_MINSOURCE 1` : Only simulate the sensor if 1 or more stars land on it
- `SIM_VISTIME 300.0` : Total exposure time of the visit.
- `SIM_NSNAP 1` : Request one snap that accounts for all `SIM_VISTIME` seconds of the simulation.

In addition, the control file `clean.params.beta` was designed to turn off many confounding effects, such as tracking errors, saturation, and blooming. Specific trim file considerations for each subtask are given below.

### C.1 Subtask 1

We started with the trim files used in W13 processing, which consist of a random star field populated by stars of SED `km50_5000.fits_g20_5140.gz`, and covering a range of magnitudes  $19 < r < 21$ . Observations were designed to be at a zenith distance of  $20.2^\circ$ , with a boresight pointing of (Ra, Dec) =  $(79.68926^\circ, -9.70229^\circ)$ . Observations were simulated in the *i*-band. These trim files are stored in the `sims3/` directory in this repository.

## C.2 Subtask 2

We chose spectrum `km20_6000.fits_g30_6020.gz` for our reference GOV-star, and replaced 80% of the objects in the original trim file with this SED. We then selected the bluest object contained in the base catalog, `kp01_9750.fits_g45_9830.gz`, for every tenth object starting at object #0, and the most populous red object in the catalog `m2.0Full.dat.gz` for every tenth object starting at object #5. These correspond to spectral types A0V and M2.0V, respectively.

Because the magnitudes in the trim files represent a 500nm magnitude (approximate *g*-band), color corrections to the requested magnitudes of the objects were required to preserve the relative brightness distribution of the as-simulated, multi-SED sources. This was done using their respective *i*-band magnitudes (i.e. a correction of  $-2.5 \log_{10} (f_{i, \text{SED}} / f_{i, \text{GOV}})$ ). Accordingly, the brightness distributions in the *gr*-bands were not optimal (this was fixed in subtask 3). The colors of each spectral type were calculated using `python/colors.py`. Script `makeNewTrims.py` was used to make these modifications to the `sims3/trim` files, yielding `sims5/trim` files (*i*-band). The only modifications made to the input trim files to simulate images in the *gr*-bands were to change the `Opsim_filter` field to indicate  $gri = 123$ ; these are stored in the `sims5gr/` directory.

## C.3 Subtask 3

First, to have similar brightness distributions in all data, we make a SED-dependent *and* passband-dependent magnitude correction to the trim files used for this task. These corrections are made in script `generateTrimFilesDcr.py`, which ran on the *i*-band trim files from `sims5`.

Second, the original observations do not pass through zenith on the given night of simulated observations. To design the observations to pass through zenith at the LSST site, which is at southern latitude  $-29.67^\circ$ , we first subtracted  $19.96437^\circ$  from all coordinates in the input trim files so that the field is centered at a declination equal to the Southern latitude of the site. This includes both the per-object coordinates and the boresight pointing `Unrefracted.Dec` in the trim file header.

We simulated this star field throughout the night of MJD 51130 (the night of the fiducial image simulations) to establish the times at which it was at an altitude of 40 and 60 degrees (zenith distance of 50 and 30 degrees; airmass of 1.55 and 1.16), and when it was closest to zenith. We chose 5 specific times at which to simulate the images: before meridian crossing at airmass 1.55; before meridian crossing at airmass 1.16; closest to zenith; after meridian crossing at airmass 1.16; after meridian crossing at airmass 1.55).

Because the instance catalog inputs to `phoSim` form an overcomplete set (e.g. the user specifies Ra, Dec, altitude, azimuth, and time of observation), it is possible to request a simulation configuration that is not physically possible. For this reason, we used

```
lsst.sims.catalogs.measures.example_utils.makeObsParamsRaDecSky subroutine  
makeObsParamsRaDecSky to generate a self-consistent set of observation parameters given the Ra  
and Dec of the field, and the times of observation established above.
```

In addition, this script synchronized the values of `Opsim_rotskypos` and `Opsim_rottelpos`. `Rotskypos` sets the orientation of Ra and Dec of the catalog, effectively a rotation of the Pole away from the Zenith. When `rotskypos=0` the Ra direction is “up” in the image, along the y-axis. This value was kept the same for all images, so that the star field was always rendered in approximately the same orientation. `Rottelpos` is defined as (`rotskypos`-180+parallactic angle), such that the angle of DCR should be along this axis. This will also be the angle of increasing Altitude in the Az,Alt coordinate system. This value is observation dependent, and was determined using `makeObsParamsRaDecSky`. Trim files for this run are found in the `sims8/` directory of this

repository.

## D Refraction and DCR Overlays

We provide here additional figures here that demonstrate the amplitude of the refraction, of the differential refraction, and the airmass and passband-dependent quality of the difference images.

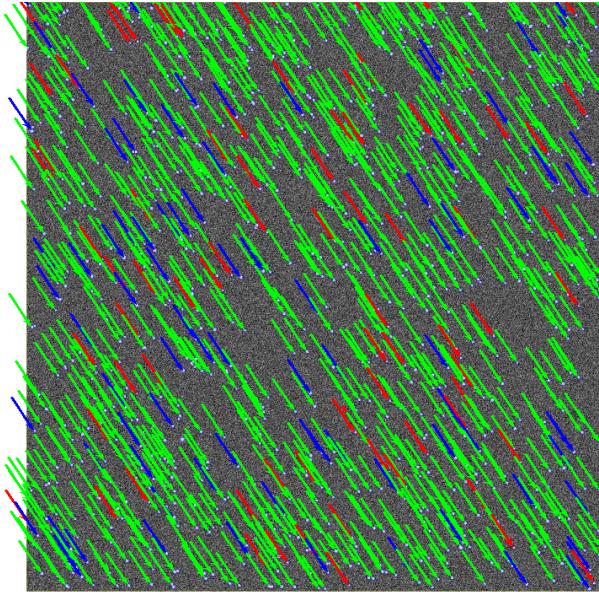
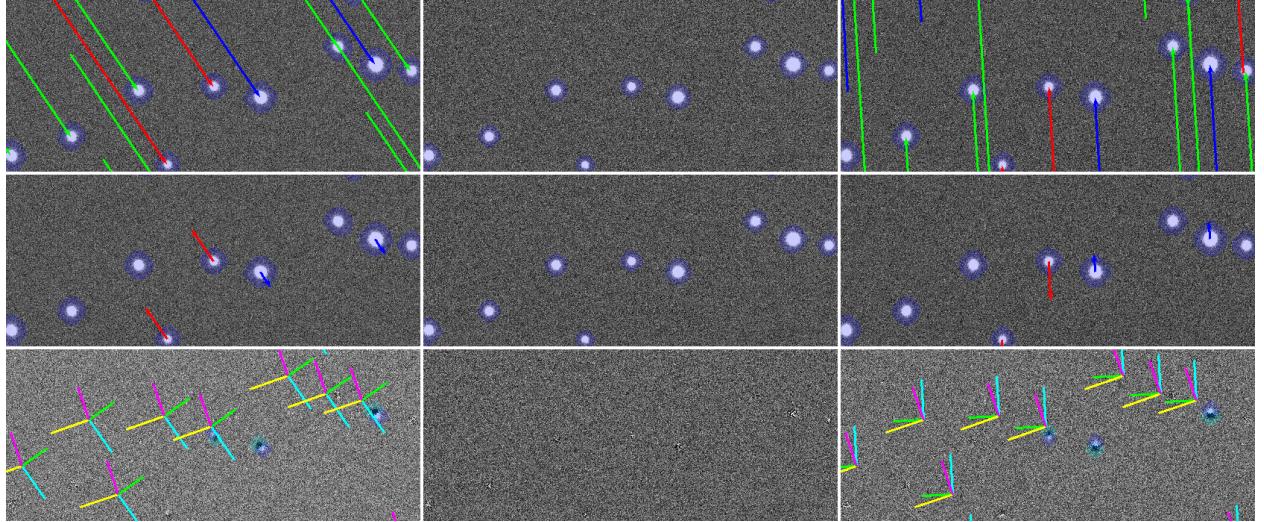
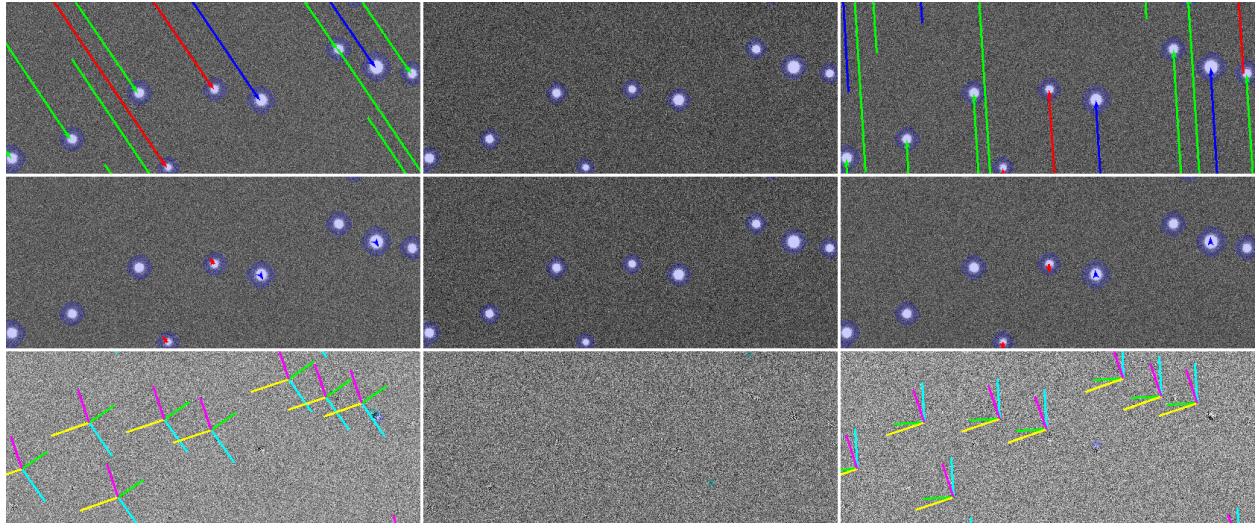


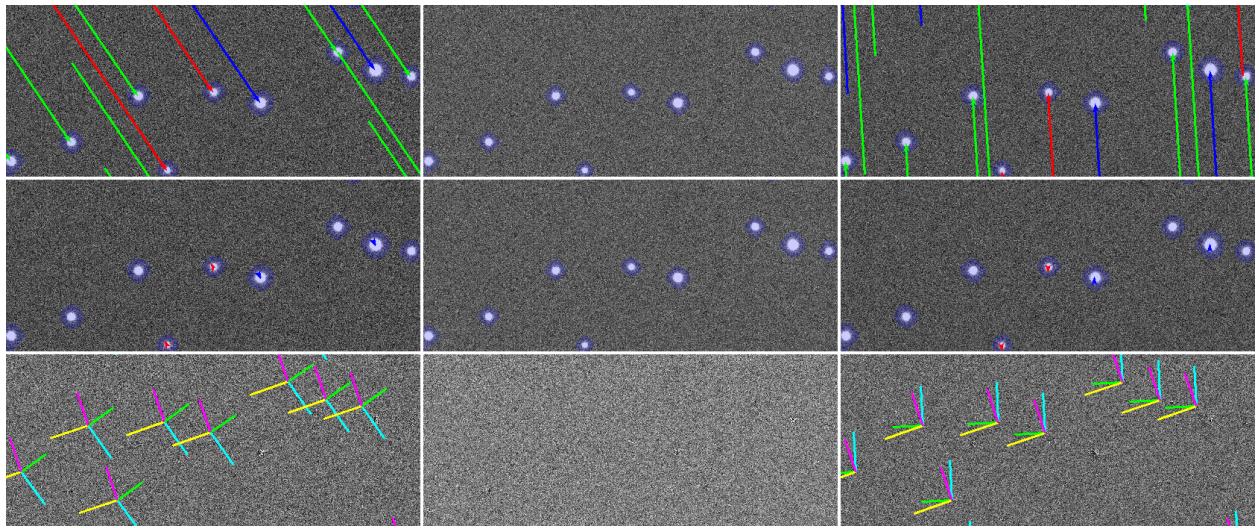
Figure 11: **Refraction:** This figure depicts the amplitude and orientation of the refraction vector in the visit A, `raft=2,2 sensor=1,1 filter=g`, seeing value 2 data. The vectors point from the unrefracted locations to the realized locations in the image. The SEDs of the sources are indicated with colors blue, green and red for AOV, GOV, and M2.0V respectively. This figure was created using the script `python/compareDcrFromSims.py`.



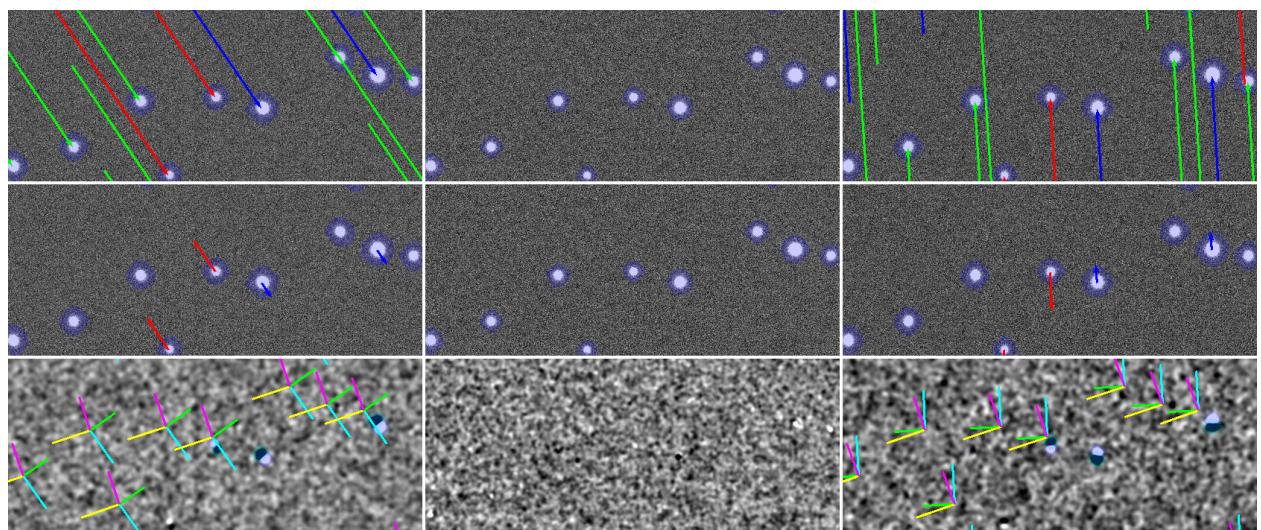
**Figure 12: Differential Chromatic Refraction and Difference Image Quality,  $g$ -band:** This set of panels demonstrates the amount of refraction (top row), differential refraction with respect to the GOV star (middle row), and the quality of the difference image (bottom row) for three sets of image differences. The first column uses science visit A, the middle C, and the third E. In all cases the template used was taken at zenith, i.e. visit C. The top rows effectively present the same information as Figure 11, but zoomed in on a particular cluster of stars. The second row subtracts off the green vector from all vectors. The residual refraction of the blue,red vectors represents differential chromatic refraction. These residual lengths have been multiplied by a factor of 100 for readability. Note that the blue vectors point along the vector to zenith, indicating the blue stars appear higher in the sky than their green counterparts, compared to an unrefracted observation. The red stars are not refracted as much and thus will appear lower in the sky. On the bottom row, we show the realized difference image quality. Note that the blue stars have positive lobes pointing along the direction to zenith, meaning the stars are “higher” in the sky w.r.t. the green stars when compared to the zenith template, while the dipoles of the red stars have the opposite polarity. For completeness, the Wcs-derived orientation of the Ra,Dec and Az,Alt coordinate axes are shown in the difference image (Ra,Dec,Az,Alt are magenta,yellow,green,cyan); see Figure 6 for more detail. This figure was created using the script `python/compareDcrFromSims.py`.



**Figure 13: Differential Chromatic Refraction and Difference Image Quality,  $r$ -band:** Same as Figure 12, but for  $r$ -band data. This figure was created using the script `python/compareDcrFromSims.py`.



**Figure 14: Differential Chromatic Refraction and Difference Image Quality,  $i$ -band:** Same as Figure 12, but for  $i$ -band data. This figure was created using the script `python/compareDcrFromSims.py`.



**Figure 15: Differential Chromatic Refraction and Difference Image Quality, Prefiltering:**  
Same as Figure 12, but using prefiltering of the science image with its Psf. This figure was created using the script `python/compareDcrFromSims.py`.