Report on Late Winter2013 Production: Image Differencing

Andrew Becker (UW), Simon Krughoff (UW), Andrew Connolly (UW),
Russell Owen (UW), Paul Price (Princeton), Robyn Allsman (LSSTC),
Yusra Al Sayyad (UW), Jacek Becla (SLAC), Steve Bickerton (Kavli IPMU),
Jim Bosch (Princeton), Scott Daniel (UW), Greg Daues (NCSA),
Gregory Dubois-Felsmann (SLAC), Mike Freemon (NCSA),
Perry Gee (UC Davis), Bill Glick (NCSA),Steve Groom (IPAC),
Joshua Hoblitt (NOAO), Mario Juric (LSSTC), Jeff Kantor (LSSTC),
Ron Lambert (NOAO), Dustin Lang (CMU), Kian-Tat Lim (SLAC),
Robert Lupton (Princeton), Bruce Mather (NCSA), Serge Monkewitz (IPAC),
Knut Olsen (NOAO), Steve Pietrowicz (NCSA), Ray Plante (NCSA),
Dick Shaw (NOAO), Douglas Smith (SLAC), Schuyler Van Dyk (IPAC),
and Daniel Wang (SLAC)
for LSST Data Management

April 9, 2013

We provide here a technical report on the LSST Data Management late–Winter 2013 production runs. This production focused on the evaluation of LSST DM image differencing code and, in particular, the rate of false detections the pipeline generates. As inputs, we used a suite of simulated images generated in semi-idealized (stars only, no saturation, cosmic rays, or defects) conditions and containing no intrinsic variability. Thus *any* sources detected in these image differences are "false positives".

We investigated two distinct operational modes of the pipeline. The first pre–filters the science image with its own PSF, yielding a point–source maximum likelihood image. The template is then PSF–matched to this image, subtracted from it, and detection and measurement happen directly on this difference. The second mode matches the template image to the science image, creates their difference, post–filters the difference with the science image PSF, and then runs detection on this maximum likelihood image. Measurement happens on the detected footprints in the difference image.

We investigated the complexity of the basis functions used to model the PSF–matching kernel, and the degree to which this kernel varies across the images. We generate several goodness of fit statistics for a control sample of sources that were *not* used in the kernel fit (notably the mean squared error), in order to assess the interpolation ability of the matching kernel. These metrics are shown to correlate with the numbers of false positives. We calculate a mean squared error metric that shows a minimum value close to the configurations that yielded the lowest number of false positives, and shows promise as a means to algorithmically choose the optimal pipeline configuration.

We compared the measured numbers of false detections to the number we expect due to random fluctuations in a background Gaussian field. The latter is a function of both seeing in the images,

and of the detection threshold used. At the fiducial detection threshold of 5–sigma, the number of expected false positives per LSST sensor is 3–12, depending upon seeing (see Section 5.1). We measure a similar number in the difference images, indicating that we are able to achieve the systematic floor in false detection rate. We examined our measured false detection rate as a function of detection threshold and find that it scales as expected. From these measures, we estimate that in order for there to be less than 1 expected false detection per sensor (i.e. ∼20% of the alert production rate) we must operate at a detection threshold of 5.5–sigma (for seeings as good as 0.6"). This result is seeing dependent.

We found a significant dependence of the false positive rate on our understanding of the variance; even a 2% misestimate of the noise may change the false positive rate by a factor of 1.5–2 at 5–sigma. Pre–filtering is preferred in this regard, as our variance planes are shown to track the variance to higher fidelity at the detection stage. We find that over/under subtraction of the image background leads to a bias in the positive to negative false detection ratio, but not on the overall number of detections. Pre–filtering may obviate the need for deconvolution of the template image when the science image has better seeing, which is a clear advantage of the technique.

We outline several additional steps we will pursue on the Summer 2013 timescale, including operating on images with more astrophysical and atmospheric complexity, improving measurement on PSF–filtered sources, adopting Gaussian processes for modeling of the spatial kernel, and using statistics calculated during the image subtraction to tune the algorithm "on the fly" (i.e. predict the numbers of false positives it will generate).

Please address any questions about this document to LSST-data@lsstcorp.org.

Contents

1 Production Scope and Goals

The focus of this production was to optimize and quantify the false detection rate in the PSF–matched subtraction of a science image and a reference image (referred to as the template image below). We used simulated images generated using semi-ideal conditions, and with no intrinsic variability in the sources. The metric we optimized is the number of false detections that are detected and measured on the image difference. As the input were simulated with no variability, any detections found in the difference images are "false positives". We calculated several additional quality metrics during the image subtraction process, and correlated them with the number of false detections, as a means to predict the number of false detections we may expect downstream from the image subtraction. Once we found an optimum configuration, we perturbed the solution along multiple dimensions to understand how the false detection rate responded.

1.1 Algorithm and Background

We outline the image subtraction process below. We assume that science image $S(x,y)$ can be modeled as a convolution of the template image $T(x,y)$ by a PSF–matching kernel $K(u,v;x,y)$ (indices u,v indicate that the kernel itself is a 2–dimensional function, which varies as a function of position x,y in the image. During convolution and correlation there is an implicit summation over u,v). The images will have different point spread functions (PSFs), which are the time–averaged transfer functions of a point source through the Earth's atmosphere, telescope optics, and into the silicon of the detector before being read out. The essence of image subtraction is to match the PSFs of these two images so that they may be subtracted pixel by pixel. We further assume that the PSF–matching kernel may be decomposed using a set of basis functions $K(u,v) = \sum_i a_i K_i(u,v)$, where the coefficients in front of each basis are determined through:

$$
\begin{aligned}
C_i &\equiv (K_i \otimes T) &&\text{(1)}\\
b_i &= \sum_{x,y} \frac{C_i(x,y)S(x,y)}{\sigma^2(x,y)}\\
M_{ij} &= \sum_{x,y} \frac{C_i(x,y)C_j(x,y)}{\sigma^2(x,y)}\\
a_i &= M_{ij}^{-1}b_j
\end{aligned}
$$

where $\sigma^2(x,y)$ is the per–pixel variance stored in the `variance` plane of each LSST `exposure`. To generate a spatially varying model for the kernel, we assume that the relative weights of the basis functions a_i themselves vary spatially, i.e. $K(u,v;x,y) = \sum_i a_i(x,y)K_i(u,v)$. We also assume that there is a differential background between the two images $b(x,y)$ that may be fit for using a low–order Chebyshev polynomial. The image difference is then calculated through $D(x,y) = S(x,y) - T(x,y) \otimes K(u,v;x,y) - b(x,y)$.

The basis functions themselves $K_i(u,v)$ are a degree of freedom in this problem. The original implementations (Alard & Lupton, 1998; Alard, 2000) used a set of 3 Gaussians (represented by our `Config` variable nGauss), each with a different width (sigGauss), and each modified by a Laguerre polynomial to a given order (degGauss). Subsequent studies (e.g. Israel et al., 2007) have suggested

5

that a constant ratio be maintained between the different Gaussian widths, such that $\sigma_{i+1} = \beta \times \sigma_i$. We use the value $\beta = 2.0$ for this production. We set the overall scale for sigma by noting that, under the assumption that the PSFs of the images are Gaussian (σ_S for the science image and σ_T for the template image), the sigma of the matching kernel should be simply $\sigma_K^2 = \sigma_S^2 - \sigma_T^2$. We use this width for the central Gaussian in the basis sequences where we use more than one Gaussian, or the sole Gaussian in the sequences when we use only one.

Detection on the difference image occurs through correlation of $D(x, y)$ with the science image's PSF, yielding optimally filtered detection image $D'(x, y) = D(x, y) \circ PSF_S(u, v; x, y)$ where $\circ$ denotes correlation (currently the DM stack uses convolution instead of correlation, reported as ticket #2552). The values of the pixels in $D'(x, y)$ provide a maximum likelihood estimate of there being a point source at that position. Detection occurs by simply finding pixels that are more than N sigma above the square root of the per–pixel variance.

In this production, we investigated two orders of operations for the optimal filtering by the PSF. In the first method, called pre–filtering, we PSF–match the template image to a pre–filtered science image:

$$
\begin{aligned}
D_{Pre}(x, y) &= S(x, y) \circ PSF_S(u, v; x, y) - T(x, y) \otimes K(u, v; x, y) \\
D'_{Pre}(x, y) &= D_{Pre}(x, y).
\end{aligned}
$$

By pre–smoothing the image that we are matching the template to – $S(x, y) \circ PSF_S(u, v; x, y)$ – we more frequently avoid the case of PSF–matching *deconvolution*, where the full width at half maximum (FWHM) of T is broader than S. This pre–filtering increases σ_S by a factor of $\sqrt{2}$. The PSF matching kernels will need to be correspondingly larger to account for the larger effective PSFs. We are able to run detection directly on this image with different measurement algorithms (see Section 3.2).

In the second, which is the classical post–filtering implementation of the technique, we create a difference image that is then post–filtered with the PSF for detection:

$$
\begin{aligned}
D_{Post}(x, y) &= S(x, y) - T(x, y) \otimes K(u, v; x, y) \\
D'_{Post}(x, y) &= D_{Post}(x, y) \circ PSF_S(u, v; x, y).
\end{aligned}
$$

2 Input Data

Specialized simulations were developed with constrained astrophysical complexity for the initial test data. The purpose of these images was to define a baseline for how well image subtraction would perform in the absence of chromatic effects (e.g. differential chromatic refractions), and perturbations in the optical system. To accomplish this the data comprised i-band simulations of a random distribution of stars with a density of ~ 1000 stars per CCD and with an equivalent r-band magnitude range ($19 < r < 21$). All stars in this sample had identical spectral energy distributions (km50_5000.fits_g20_5140.gz; see the phosim page,http://ls.st/1vo for links to the spectra), and, for all sources, variability, proper motion, and parallax were set to zero. All simulations were made at the zenith.

Nine sensors of the central raft were simulated, for a total of 27 difference images per configuration to use in our comparative analysis. In each case a pair of snap exposures was generated, however

only the first snap was used in the analysis. Each simulation run was generated using the following configurations:

- Visit 6866600: A pair of 300s exposure template images with an atmospheric seeing value of 0.88 arcsec.

- Visit 6866601: A pair of 15s exposure exposures with 0.6 arcsec atmospheric seeing.

- Visit 6866602: A pair of 15s exposure exposures with 0.88 arcsec atmospheric seeing.

- Visit 6866603: A pair of 15s exposure exposures with 1.2 arcsec atmospheric seeing.

For each configuration, v3.2.5 of the LSST phosim package (see `http://ls.st/1vo` for usage) was used to generate the data. The sky background was dark and set to produce ~150 counts per pixel, with airglow variations set to zero. Telescope and camera perturbations, tracking errors, field rotation, and saturation and blooming effects were all disabled. To enable the use of eimages directly within the analysis (i.e. to remove the need for the construction of a full CCD from the amplifier images, calibration of the simulated data, and the generation of photometric zero points) the pixel quantum efficiency variations were set to zero, cloud screens were turned off and no cosmic rays were generated. The phosim commands file for the generation of the simulated data is listed in Appendix A. Reference catalogs and astrometry.net indices were constructed from the input catalogs. A secondary set of simulation images was generated for an extended Summer 2013 analysis. These data are described in Appendix B.

3   Work Performed

Major tasks that were completed in the run–up to production include: implementing pre–filtering for image subtraction (ticket #2551); implementing source selection for image subtraction (#2554, #2596); implementing measurement for difference image source (`DiaSource`) characterization (#2595); implementing RegisterTask for image–to–image registration (#2562, #2650); defining and implementing metrics for testing ip_diffim (#2636); designing a tool to scrape metadata from data repositories (#2640); and implementing flux measurement for PSF–filtered sources (#2754).

Below we describe in detail the overall pipeline that was established as the results of these tasks. We follow with in–depth analysis of implemented algorithms that are central to this production.

3.1   Pipeline Implementation

At the beginning of the pipeline, we have as an input the science image $S(x, y)$ as a `calexp` instance. The bounding box of the science image, in celestial coordinates, is ideally used to query the coadd repository to return the coadd patches that overlap the image. However, because of difficulties in fitting absolute astrometry to the images (Section 11.1), we instead took the approach of directly using the `calexp` of the deep visit (v6866600) as the reference visit (as opposed to taking this deep `calexp` and turning it into a coadd).

Next, we used the DiaCatalogSourceSelector to query the ImSim reference catalog for appropriate sources to use for PSF matching. This selector allows the user to specify the brightness and color range of the objects, toggle star or galaxy selection, and to include variable objects or not. In this semi-ideal case we use all sources. We selected a fraction (1/5) of these objects to serve as a control sample, in order to assess the effectiveness of $K(u, v; x, y)$ to subtract objects that were *not* used in the least squares fit.

The reference and science `calexps` were then astrometrically registered to each other, with the resampling operations happening on the reference visit as it has the larger signal–to–noise (S/N). Object–to–object matching, done using the source list above cross–matched with a source list extracted from the coadd, was used to provide a relative astrometric transformation between the two images. RegisterTask was used to do the resampling. The resulting warped reference image was what we use as the template image $T(x, y)$. The residuals of the object–to–object match were persisted as part of the debugging infrastructure. We assumed that the background subtraction on the input `calexps` removed the high order features in the backgrounds; therefore we simultaneously fit for a 1st–order spatially varying differential background $b(x, y)$ along with the kernel coefficients.

Images $S(x, y)$ (optionally pre-filtered with $PSF_S(u, v; x, y)$) and $T(x, y)$ were sent to ImagePsf-MatchTask, whose purpose is to fit for $K(u, v; x, y)$ and to produce the difference image $D(x, y)$.
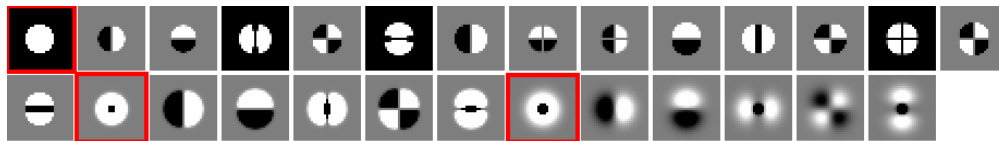

Figure 1: Example basis set with nGauss=3 (highlighted in red), sigGauss=(0.5 sig, 1.0 sig, 2.0 sig), and degGauss=4, yielding Laguerre polynomials of order (4,2,2).

The first step is to use the PSF Gaussian widths to define the *sizes* of the Gaussians' shapes in $K_i(u, v)$. As mentioned above, the main width was chosen to be $\sqrt{\sigma_S^2 - \sigma_T^2}$. We investigated variations on the configuration where we used one Gaussian with the above width, and where we used three Gaussians, the central of which has this width and the others being a factor of $\beta = 2.0$ smaller and larger. We modified each of the Gaussians by a set of Laguerre polynomials of a given order. The smallest Gaussian was modified to the specified order, with the others modified by floor(order/2). Figure 1 provides an example basis set with this configuration. The total number of shapes in each basis were: $\sum_i^{\mathrm{nGauss}} (\mathrm{degGauss}_i + 1) \times (\mathrm{degGauss}_i + 2)/2$ [¶]. The dimensions of the PSF matching kernel were chosen to be 6 times the largest Gaussian width, with a minimum

---

[¶]A notable exception to this algorithm is found in the case where $\sigma_T > \sigma_S$, i.e. a sharpening must be done to the template to match it to the science image. This is the case of deconvolution. A theoretical prescription for basis design in this case is given in Ulmer (2010). In our implementation of this algorithm, nGauss is *always* fixed to value 3, and degGauss is always fixed to 3. The widths of the Gaussians are determined through the algorithm specified in Ulmer (2010), using as inputs the sequence of Gaussians that we would have used to match a Gaussian of width σ_S to σ_T (i.e. as if we would have convolved the science image and not the template image). There are a variety of reasons that actually operating on the science image is undesirable, including: the science image is lower S/N than the template; the science image is likely to have masked defects, including cosmic rays and bad pixels, whose reach would be expanded due to the convolution; unmasked features such as faint cosmic rays are more likely to look PSF–like after convolution (cosmic rays will likely have been sigma clipped out of the coadd). However, it is not clear that deconvolving the template image is in principle any better. A far better option, which we explore here, is to pre–filter the science image, which will in the vast majority of cases provide a larger PSF (by a factor of $\sqrt{2}$) to match the

kernelSize of 21x21 pixels. For kernels that are significantly smaller than this, the Gaussians have significant (non–zero) power at the kernel boundaries, leading to square systematic artifacts at the scale of the kernel in the difference images (see Figure 2).
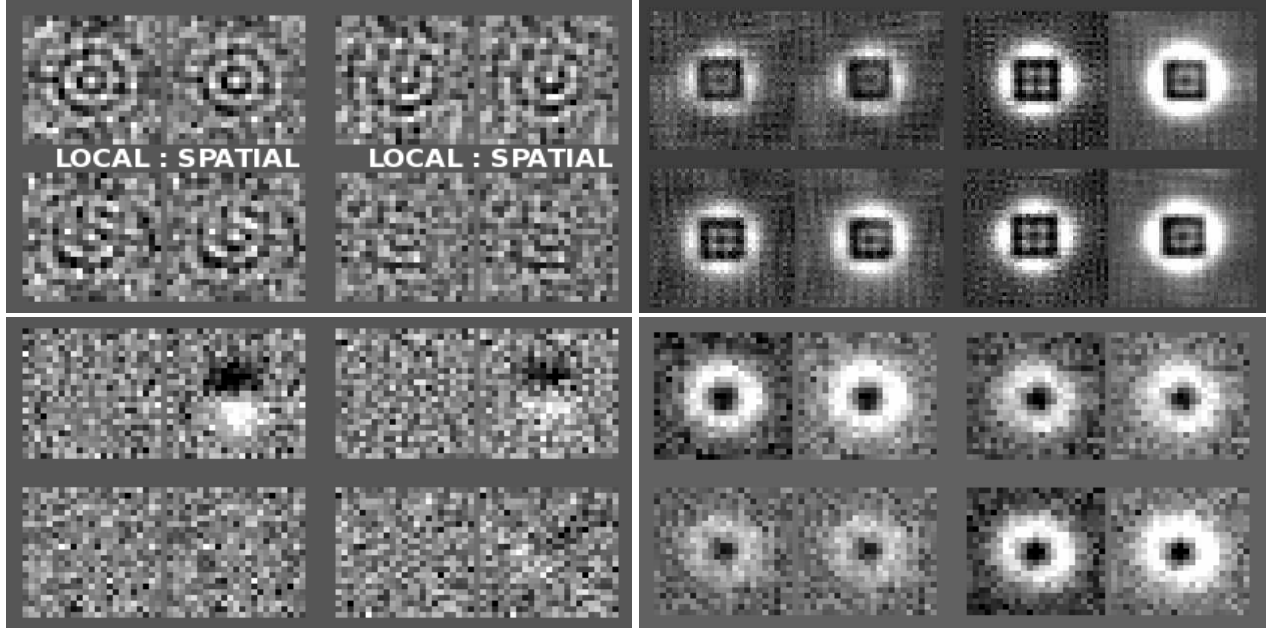

Figure 2: Four example panels showing failure modes of the image subtraction software. Each panel itself shows a 2x2 pair of `KernelCandidate` difference images; on the *left* subpanel is the residual from the LOCAL fitting, and on the *right* from the SPATIAL fitting. Starting in the upper left and going clockwise, the first panel shows the residuals when deconvolving the template image; the ringing is characteristic of the deconvolution process. Next, a set of difference images where the kernel dimensions (kernelSize) are too small to match the sources, leaving square kernel–sized residuals around each object. Next, a set of images where the kernel shape (sigGauss) is inappropriate to match the sources; note that the LOCAL residuals show circular residuals, indicating the basis set itself is at fault (as opposed to the spatial model). Finally, in the lower–left, a set of images that demonstrate how the residuals degrade when the spatial kernel model is at fault. Note that the LOCAL residuals appear to be white noise, while the SPATIAL residuals show a clear dipole signature. For this reason, a comparison between the LOCAL and SPATIAL residuals is a useful diagnostic of the spatial model.

For each source j returned by DiaCatalogSourceSelector, we created a `KernelCandidate` instance, which holds substamps of $T_j(x, y)$ and $S_j(x, y)$ centered on the source. These were used to fit for a local solution $K_j(u, v)$; the ensemble of local solutions at each source's x, y was used to fit for the full spatial solution $K(u, v; x, y)$. We emphasize that the dimensions of the substamps are important. Note from equation 2 that the coefficients a_i are derived from a convolution of the template substamp with the kernel basis functions K_i. This convolution means that kernelSize//2

---

template to. This is the case for v6866601 in this study, where the post–filtered analysis yields deconvolution of the template, while the pre–filtered analysis does not.

pixels along the border of each substamp are rendered unusable. To maintain a significant number of pixels in C_i, we set the `KernelCandidate` dimensions to be twice that of the kernel (i.e. the number of pixels remaining in C_i is equal to the number of pixels in the kernel, which is fewer than the number of bases i). For each `KernelCandidate`, we solved for $K_j(u,v)$ and create a local difference image $D_j(x,y)$. We evaluated several statistics on this local difference image, normalized by the square root of its variance, which puts the pixels in units of σ (see also Section 3.3). We measured the mean sigma, the RMS of this distribution, the χ^2 of these pixels normalized by the degrees of freedom (number of pixels in $D_j(x,y)$ minus the number of kernel bases), and the mean squared error. These are referred to as the LOCAL kernel metrics, and reflect how appropriate the chosen kernel basis is (see again Figure 2).

The ensemble of `KernelCandidates` was then used to constrain a spatial model of the kernel $K(u,v;x,y)$. This was done using the `SpatialCellSet` formalism, whereby the `KernelCandidates` are distributed across the image in a grid of `SpatialCells`. This formalism attempts to distribute constraints on the spatial model evenly across the image. At most 3 `KernelCandidates` per `SpatialCell` were input to the spatial kernel fit. For the spatial kernel model, we assumed that each of the kernel coefficients a_i may be represented by a Nth order 2–dimensional Chebyshev. Normally, we would perform sigma–clipping iterations on candidates having bad residuals after spatial modeling. However in this production we expect that all objects should fit, so we performed no sigma clipping. Having generated the full spatial solution $K(u,v;x,y) = \sum_i a_i(x,y)K_i(u,v)$, we evaluated the kernel solution at the position of each `KernelCandidate`, created a difference image using this kernel, and recalculated the metrics defined above. These are referred to as the SPATIAL kernel metrics. Importantly, we also interpolated this solution to the positions of the control sample, which were explicitly *not* used in the spatial fit, and evaluate these same metrics. The differences between the `KernelCandidate` LOCAL and SPATIAL metrics, as well as the values of the SPATIAL metrics for the control sample, reflect the appropriateness of the spatial model and our ability to interpolate and extrapolate to the full extent of the images. We used the Mean Square Error (MSE) of the control sample to examine the tradeoff between bias and variance in the spatial interpolation.

We next ran the SourceDetectionTask on the difference image, using a "polarity" of *both* to search for both positive and negative deviations. We initially used a nominal detection threshold of 5 σ. In the case of pre–filtering, we detected on the difference image directly. In post–filtering, we convolved the difference image with its PSF (by design, the same as the PSF of S) before detection. Since the fundamental image operations are the same, the images immediately preceding the detection step (D') should be *exactly* the same. Detections were merged using a grow radius of 2 pixels.

Finally, we ran DipoleMeasurementTask on the detections to create `DiaSources`. DipoleMeasurementTask is an extension of SourceMeasurementTask that includes specific dipole measurement algorithms. Measurement was necessarily different in the pre–filtered and post–filtered data. In particular, in post–filtering, detection happens on the filtered maximum likelihood image and measurement on the associated pixels in the unfiltered difference image. In pre–filtering, the latter image does not exist, and thus detection and measurement happen on the same image. This influences how measurements made on the sources are interpreted (see Section 3.2). While these measurements were straightforward to implement, additional work needs to be done on how to implement the remainder of the measurement suite on post–filtered data (e.g. shapes, moments, and PSF goodness–of–fit metrics).

10

All `DiaSources` were associated with the `calexp`'s sources and with the reference catalog, with a matching radius of 3". `DiaSources` that matched with sources may be true residuals around stars *or* true false positives that are also found in the `calexp` SourceDetectionTask. For this reason, we used the associations with the reference catalog objects as the assessment of if a given `DiaSource` matched with an input object or whether it was an orphan (e.g. a statistical fluctuation in the background). In addition, all `KernelCandidates` were persisted with the above `LOCAL` and `SPATIAL` metrics. Repositories were accessed with a specially written RepositoryIterator that aggregates `KernelCandidates` from a given source across multiple output repositories. Finally, the `DiaSource` lists for all production repositories were compared to find the configurations that yielded the lowest numbers of false detections.

3.2 Improvements to Measurement

We measured the flux of `DiaSources` on post–filtered difference images using PSF–weighted flux photometry (as implemented in the LSST measurement algorithm "flux.psf"). For measuring fluxes of point sources on pre–filtered images we implemented a new algorithm "flux.filtered".

We start with the definition of PSF–weighted flux assuming a normalized psf, ϕ:

$$flux = \frac{\sum_{(x,y)} I(x,y) * \phi(x,y)}{\sum_{(u,v)} \phi^2(u,v)}$$

where $I(x,y)$ is the image before PSF filtering and $\phi(x,y)$ is an image of the PSF centered on the source. This equation assumes an isolated point source, an assumption that is reasonable for difference images. We do not weight by the per–pixel variance to avoid non–linearity in the flux measurement, at the expense of less precise measurements at the bright end.

Filtering is performed by convolving the image with its PSF, so the peak of the filtered image of a given source is:

$$max(I_{filt}) = \sum_{(x,y)} I(x,y) * \phi(x,y).$$

Hence the equation for measuring its flux reduces to:

$$flux = \frac{max(I_{filt})}{\sum_{(x',y')} \phi^2(x',y')}.$$

We further define

$$w(x,y) \equiv \frac{\phi(x,y)}{\sum_{(x',y')} \phi^2(x',y')}.$$

To calculate the variance, we assume a point source model for the intensity

$$I(x,y) = A\phi(x,y) + \epsilon(x,y)$$

$$\langle flux \rangle = \langle \sum_{(x,y)} w(x,y)(A\phi(x,y) + \epsilon(x,y)) \rangle$$

with a white noise contribution $\epsilon(x, y)$. Since $\langle\epsilon(x, y)\rangle = 0$ and $\langle\epsilon(x, y) * \epsilon(x', y')\rangle = \sigma^2(x, y)\delta(x - x', y - y')$:

$$\langle flux \rangle = \sum_{(x,y)} w(x, y)A\phi(x, y)$$

and

$$Var(flux) \equiv \langle (flux - \langle flux \rangle)^2 \rangle$$

$$Var(flux) = \left\langle \left(\sum_{(x,y)} w(x, y)\epsilon(x, y) \right)^2 \right\rangle$$

$$Var(flux) = \sum_{(x,y)} w^2(x, y)\sigma^2(x, y).$$

Substituting $w(x, y)$

$$Var(flux) = \frac{\sum_{(x,y)} \phi^2(x, y)\sigma^2(x, y)}{(\sum_{(x',y')} \phi^2(x, y))^2}.$$

This implies that the error σ_{filt} on the flux at the location of $max(I_{filt})$ is

$$\sigma_{filt} = \frac{\sqrt{(\sum_{(x,y)} \phi^2(x, y)\sigma^2(x, y))}}{\sum_{(x',y')} \phi^2(x, y)}.$$

The numerator evaluates to the square root of the value in the variance plane at the location of the peak flux.

The center of each source is determined using an LSST adaptation of the SDSS centroiding algorithm "centroid.sdss". In general the source will not be centered on a pixel, so "flux.filtered" uses a Lanczos 4 resampling kernel to shift the source to be centered on the nearest pixel before applying the equations above.

3.3   KernelCandidate Statistics

We implemented several diagnostics on the quality of the image subtractions for a given `KernelCandidate`. This was done for the original $T_j(x, y)$ to $S_j(x, y)$ kernel fit, yielding LOCAL quality statistics. These were re–evaluated using the spatial kernel model evaluated at the `KernelCandidate` location $K(u, v; x_j, y_j)$, yielding SPATIAL quality statistics. The task metadata was augmented with summary statistics of these values (mean, median, RMS), and the information on each `KernelCandidate` persisted in a new `kernelSrc` data product. We briefly describe more useful of these statistics below.

The first statistic, KCDiffimStDev, measures the standard deviation of all the non–EDGE pixels in the difference image, normalized by the square root of the variance of the difference image. In the idealized case, this value should be 1.0, i.e. pixels in the image are drawn from a normal (0,1) distribution. In cases of overfitting or underfitting, this value will be lower than / higher than 1.0.

We also calculate a reduced χ^2_ν value called KCDiffimChiSq. The numerator of this term is the sum of the squares of the noise–normalized pixel values. This sum is normalized by the number of pixels in the data minus (the number of basis shapes in the kernel plus 1).

Finally in order to investigate the bias–variance trade–off in the overall fit, we calculate the mean square error (MSE) of each `KernelCandidate`'s difference image as statistic KCDiffimMseResids. We define the bias as $|data - model|$, the variance as $(data - model)^2$, and the MSE as `bias`^{2} `+ variance`. In this context, the bias is the mean of the difference image, and the variance is the mean square of the difference image. Ideally, the MSE gives a means for identifying the location in parameter space of optimal fitting without overfitting.

For all statistics, we have a FITTED sample, which are the `KernelCandidates` that were used to derive the spatial model. Importantly we also have a CONTROL sample of `KernelCandidates` that were explicitly held out of the fitting to use for cross validation.

4 Production Runs

We ran several pre–productions to tune the following aspects of the pipeline: the dimensions of the kernels and `KernelCandidates`; the widths of the Gaussians (sigGauss) based on the FWHM of the input images; and the scaling of the Gaussians (β) in the basis sequence. Figure 2 provides a rogues gallery of image subtraction failures encountered at this stage, each centered on `KernelCandidates`, and outlines reasons for each failure. Figure 3 provides a gallery showing what the results from a successful image subtraction appeared like, for post–filtering on the *top* and pre–filtering on the *bottom*.

The main production runs included: an outer loop over the use of pre–filtering or post–filtering for detection; variations of the complexity of the Kernel basis, including the number of Gaussians (nGauss = 1,3) and the degree of the modifying Laguerre polynomials (degGauss = 1..6); and the spatial order of the Chebyshev polynomials on $a_i(x, y)$ (spatialKernelOrder = 1..6). This results in $2 \times 2 \times 6 \times 6 = 144$ permutations of the configs. See the following example python code for how the loops iterate over the configs.

```
for prefilter in (True, False):
    for nGauss in (1,3):
        for degGauss in xrange(1,7):
            for spatialKernelOrder in xrange(1,7):
                runDiffimPipeline(prefilter, nGauss, degGauss, spatialKernelOrder)
```

The final production run was labeled **production10**, and data are available on the NCSA compute cluster[†].

In the post–filter analysis, matching of the template image with visit 6866601 yielded a template *deconvolution*. Matching of the template image with visit 6866602 also yields a deconvolution (the images were generated to have the same seeing) for 5/9 sensors. This provides a test of how well the pipeline is able to perform in the case of moderate deconvolution (FWHM the same to within 0.1 pixels). The matching of the template with visit 6866603 with post–filtering, and of all visits with pre–filtering, yields a convolution (smoothing) of the image.

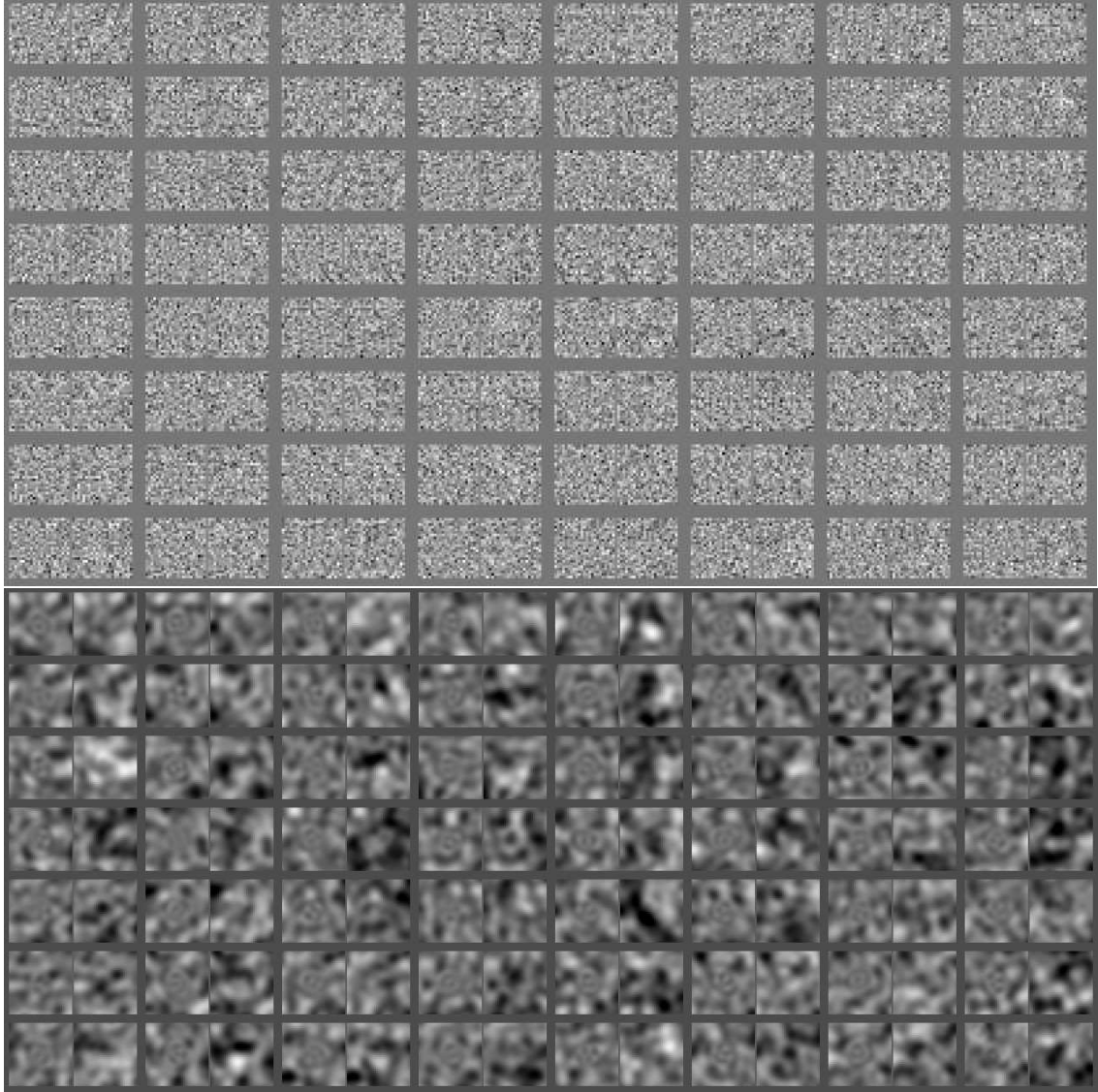[†]/nfs/lsst8/becker/sparse_diffim/production10_sparse

Figure 3: Pairs of LOCAL:SPATIAL `KernelCandidate` difference images in the case of successful image subtraction, similar to Figure 2. The *top* set of data are from post–filtered analysis, while the *lower* set are from pre–filtered data. Note that the noise is smoother in the pre–filtered data due to convolution with the PSF.

Finally, we ran several post–productions focused on perturbing the solutions that yielded the fewest numbers of `DiaSources` along specific dimensions. This included: the RMS of the WCS solution used by RegisterTask; including a DC offset in the source positions prior to RegisterTask, to determine the ability of the kernel basis to accommodate astrometric misregistration; the size of the `SpatialCells` used in the spatial interpolation, which effectively regulates the numbers of constraints that go into the spatial fit; and modifying the detection thresholds to determine how the number of false detections scales with threshold. See Section 6 for discussion of how these perturbations affect the results.

5   Results: False Detection Rate

5.1   Theoretical

We model a source–free image (i.e. with a background described by a Gaussian process) using the statistics of Gaussian random fields (Kaiser, 2004). For an image with Gaussian noise convolved with a Gaussian PSF with width σ_g the density of peaks above a given detection threshold, ν is given by

$$n(>\nu) = \frac{1}{2^{5/2}\pi^{3/2}}\nu e^{-\nu^2/2} \tag{2}$$

To estimate the number of random positive detections for a given image we must normalize this density by the total number of realizations of the PSF within the image, i.e.,

$$N_{total}(>\nu) = n(>\nu) * nrow * ncol / \sigma_g^2 \tag{3}$$

where nrow and ncol are the number of rows and columns. In difference image analysis, we expect twice this number, since we are sensitive to both positive and negative–going fluctuations.
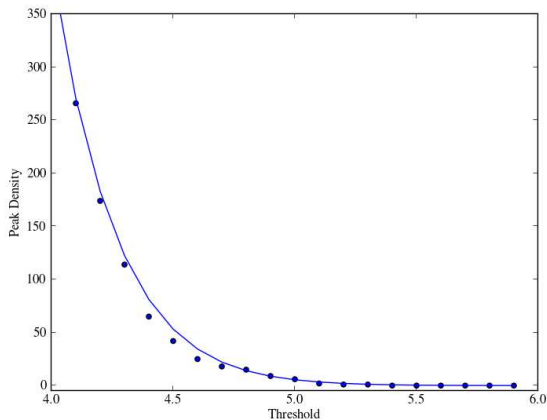

Figure 4: A comparison of the predicted number of random peaks (solid line) detected in an image, as a function of significance, with those derived from a Monte-Carlo realization of a Gaussian noise image (solid points).

15

Best Results: Production 10								
Visit	Pre–filter	N_{Expect}	$N_{Measured}$	N_{Orphan}	N_{Match}	N_{Random}	$N_{D<100}$	N_- / N_+
v6866601	True	108±27	70	60	10	3	4	1.8
	False	$\cdots$	143	51	92	2	11	0.4
v6866602	True	45±18	42	36	6	2	1	2.6
	False	$\cdots$	53	45	8	2	1	0.5
v6866603	True	27±18	23	23	0	1	0	2.8
	False	$\cdots$	36	35	1	1	0	3.4

Table 1: Total number of `DiaSources` detected at 5–sigma from all 9 sensors in raft 2,2, for the configurations that yielded the lowest number of false detections ($N_{Measured}$). N_{Expect} lists the total number we expect to detect as random fluctuations, from Table 3, and are the same for pre and post–filtering. Whether a detection is a match or an orphan (N_{Match} and N_{Orphan}) is determined using a 3" match radius with the input reference catalog. We list the number of background fluctuations that are expected to randomly associate with a Source N_{Random}, given the density of objects in the images and a 3" match radius, assuming that N_{Orphan} represents 96% of the total population of detections from Gaussian fluctuations. We also list the number of false detections that are found within the outer 100 pixels of the image $N_{D<100}$. The final column lists ratios of the number of negatively detected false detections to those with positive flux.

Figure 4 shows good agreement in a comparison of this model to a Monte-Carlo simulation for thresholds $4 < \nu < 6$ for a 4000x4072 pixel image with 0.88 arcsec seeing. The variance on these measures is calculated using 1e6 random realizations.

5.2   Empirical

After running an ensemble of 144 pipeline configurations, we aggregated the numbers of false detections. We report here the results for each of the 3 visits, and for pre– and post–filtering for detection. We cut on `flags.pixel.edge` to remove spurious detections on the immediate border of the image; other than that we do no other cuts other than requiring that the `centroid.sdss` is finite. We separated the `DiaSources` that match with the reference catalog (3" matching radius) from those that don't (referred to as orphans) to investigate the origins of the detections. We found no significant correlation of false detections rate with sensor within the raft (although the per–sensor optimal configs may be different, Section 7); therefore we report the aggregate results for all 9 sensors in the analyses that follow.

A set of "heat maps" is presented in Figure 5, that demonstrate the total number of detections across all 9 sensors as a function of the config parameters spatialKernelOrder and degGauss, for nGauss = 3. Qualitatively, these heat maps indicate that a Chebyshev spatial order greater than 3 is required for all data. In addition, with the exception of the deconvolution case (v6866601 and prefilter=False), the Gaussians need to be modified by Laguerre polynomials of order 4 or higher. The minimum of these detection maps are found using the configs of highest complexity, with a very flat slope in the number of false detections beyond spatialKernelOrder>3 and degGauss>3. There is not much evidence for an upturn in the numbers of false detections, meaning we have likely not reached the case of severe overfitting using these configs.
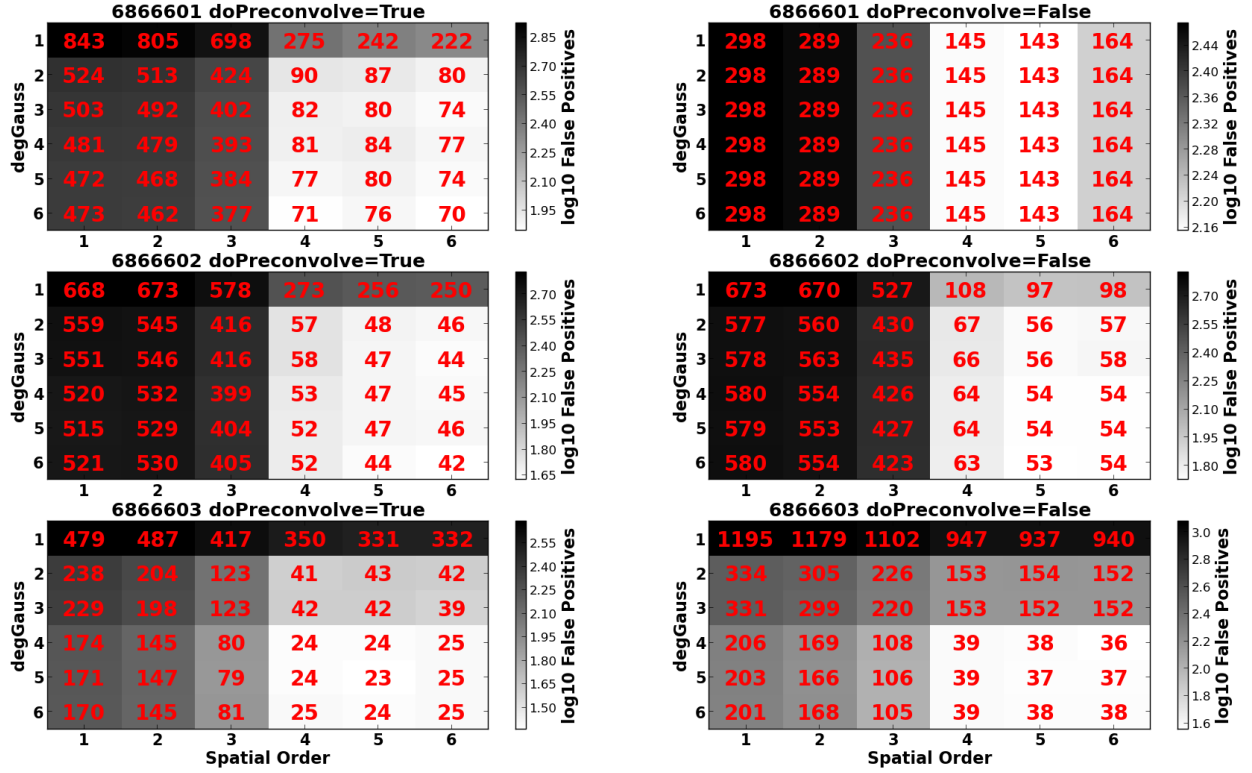
Figure 5: "Heat–map" detailing the total numbers of 5–sigma false positives as a function of two main degrees of freedom: the maximum order of basis Laguerre polynomials along the y–axis, and the spatial order of the Chebyshev expansion of the spatial model along the x–axis. The *left* column shows these results for pre–filtering, while the *right* column shows this for the post–filtered analysis. The best results listed in Table 1, with the associated configurations from Table 2, correspond to the minimum in each of the subpanels.

Best Configs: Production 10				
Visit	Pre–filter	nGauss	degGauss	spatialOrder
v6866601	True	3	6	6
	False	3	1	5
v6866602	True	3	6	6
	False	3	6	5
v6866603	True	3	5	5
	False	3	4	6

Table 2: Configurations that led to the best results listed in Table 1. All post-production runs include perturbations about these configurations.

The numbers $N_{Measured}$ of false detections in each best case are given in Table 1, with the total number expected from the Monte Carlo analysis given as N_{Expect}. We undertook a manual inspection of all difference images to associate the `DiaSources` with sources in the images, and also use a 3" association with the input source catalog to define matches and orphans. Associations are listed in the N_{Match} column of Table 1, with the unmatched false positives in the N_{Orphan} column. The former are likely to come from systematics in the subtractions of stars, while the latter would be due to fluctuations in the background. We note two effects that may impact N_{Match}. The first is that, with a 3" matching radius, 4% of random fluctuations will by–chance associate with a Source. Assuming that N_{Orphan} represents 96% of the background fluctuation population, we estimate the number of fluctuations from that same population that would have associated with Sources as N_{Random}. Finally, we count all matches that occur within 100 pixels of the image boundary, where we might expect deficiencies in the various spatial extrapolations (astrometry, backgrounds, PSF–matching kernels) to yield systematic residuals, as $N_{D<100}$. Figure 6 provides the distribution of distances from the edge of the sensor for all false detections in *blue*, with a random sample of points in the image given in *green*. Uncertainties are plotted using the square root of the number of points in each bin, and the histograms are normalized to provide a probability density. Aside from the deconvolution reduction, we find no significant correlation of false detections with the boundaries of the fitting functions. We note that we are performing better than theory for visit v6866601 and pre–filtering, although this is only 1.4 times the scatter in the expectation value. Otherwise, the results are as–expected from our simulations, indicating that we are near the systematic floor in false detection rate. We also note that there is a slightly higher number of associations with sources than we would expect randomly ($N_{Random} < N_{Match}$), although this is very small and at the rate of 1 every 4–5 images.

In the last column of Table 1 we list the ratio of negatively detected to positively detected `DiaSources`. This ratio is significantly higher than 1.0 for all pre–filtering visits, suggesting that there is an over–subtraction of the background in the image processing, which will bias this ratio. The observed ratios of 2–3 suggest that there is a background overestimation of order $1-2\%$ causing lower–sigma fluctuations to be measured at 5–sigma (Figure 7). The post–filtering ratios vary by a similar amplitude, but with a mixture of possible over– and under–subtractions.

For completeness, the optimal configurations yielding these false detection rates are given in Table 2.
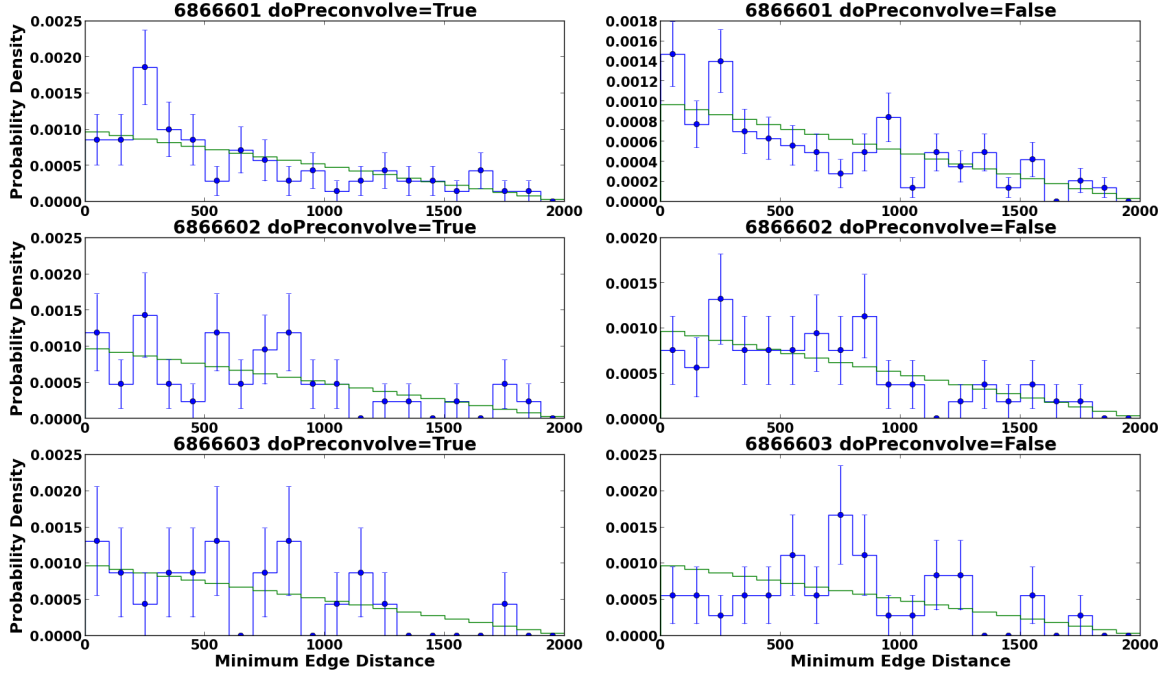
Figure 6: Distribution of false detections as a function of the minimum distance from the edge of the sensor. The *green* line shows the results from a random distribution of points. The *blue* lines show the measured distributions, normalized as a probability density ($\sum y\ \Delta x = 1$). Error bars come from the square root of the number of points in each bin.
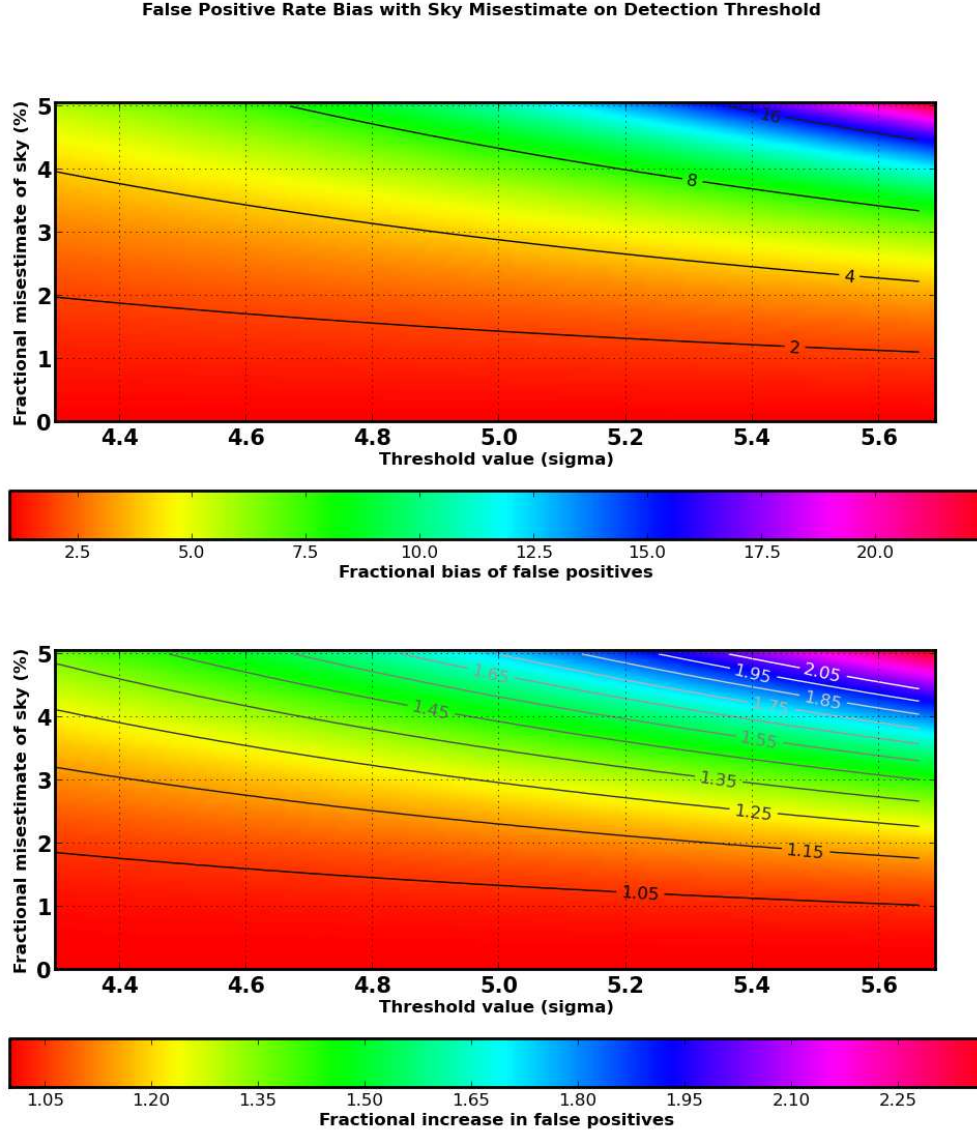
Figure 7: Misestimation of the sky value has an effect on both the number of false detections and relative abundance of positive to negative false detections. The top pane of this figure shows the relative number of +ve/-ve (-ve/+ve) detections if the sky background is under (over) subtracted at a particular fractional level. The bottom panel shows how the total number of false detections changes for the same misestimation. This is not a strong function of the seeing, so only the best seeing example is shown.

Theoretical Number of False Detections per Sensor				
Visit	FWHM (pixels)	$\sigma = 4.5$	$\sigma = 5.0$	$\sigma = 5.5$
v6866601	3.0	114±11	12±3	1±1
v6866602	4.4	54±8	5±2	0.5±0.5
v6866603	6.0	29±6	3±2	0.2±0.4

Table 3: Total number of false detections that we expect based on a 4000x4072 pixel sensor. This number is dependent on the seeing and detection threshold, so we list the values at a threshold of 4.5, 5.0, and 5.5 sigma for the seeings in the 3 visits used in this production. We list the total number of positive *plus* negative detections; they should be present in equal quantities. The variance is as determined from Monte Carlo simulations.

6   Results: False Detection Dependencies

After finding the configurations that yielded the minimal numbers of false detections, we then perturbed the solutions to examine how the numbers of false detections scaled with different effects. This included, in order of importance for this production: the difference image detection threshold; the number of KernelCandidates going into the spatial model; and astrometric registration errors. Using the theoretical analysis of the number of false detections expected in a 4k x 4k image as a function of detection threshold outlined in Section 5.1, we examined the requirements this sets on our ability to model the variance and backgrounds in the images. We compared the empirical variation of false detections with detection threshold in our data, and show below that our results are consistent with reaching the systematic floor in these data.

6.1   Effect of Uncertainty on Measured S/N

Section 5.1 discusses the theoretical foundation that sets the floor on the number of false positives we expect from a pure Gaussian random field. The number of false positives is a strong function of the seeing and the confidence level of the detection. By comparing the empirical false positive rate to the analytic predictions we can quantify how close we are getting to the theoretical floor in these data. Because the number of false positives is such a strong function of detection threshold, misestimation of the signal to noise of the detections has a large impact on how many false positives are observed.

The Figure 8 shows the effect of misestimating the noise in the detections for each of three different seeing values ($0.6''$, $0.88''$, $1.2''$ respectively). In each case the top panel shows the number of additional theoretically expected false positives in an LSST size CCD given a fractional misestimation of the noise in the image and a detection threshold (recall the baseline numbers are given in Table 3). A 1% under-estimate of the noise corresponds to 4 additional false positives at a threshold of 5σ in the best seeing case, compared to 12 expected, and 1 additional false positive in the worst seeing case, compared to 3 expected. These numbers roughly double if the noise is underestimated by 2%. The bottom panels show the effect of over-estimating the noise in the measurement. The effect is to reduce the number of false positives. A 1% under-estimate of the noise typically results in 2 fewer false positives per chip, at 5–sigma.
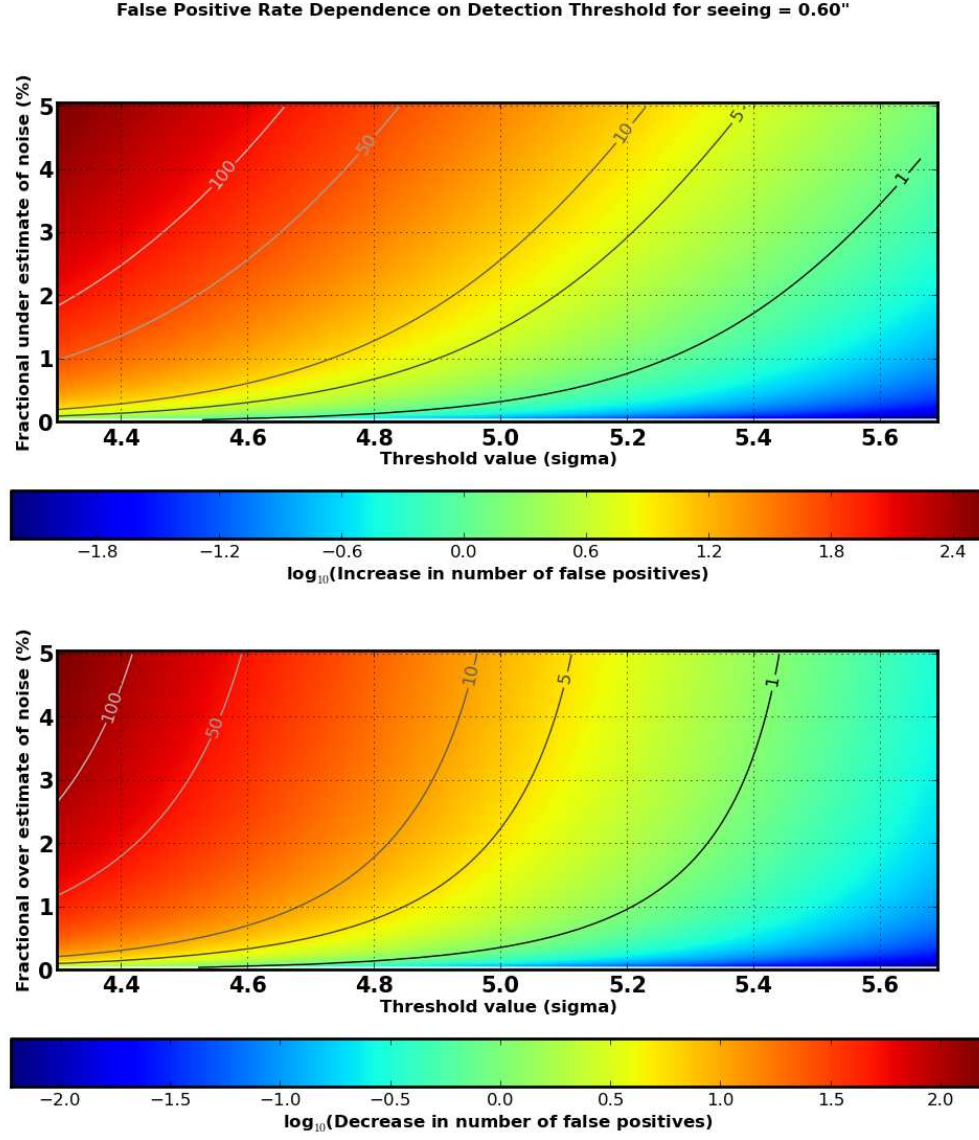
(a) 0.6″ FWHM

Figure 8: We plot here the expected change in the number of false positives as a function of detection threshold and noise misestimate. The panels on these 3 pages are for three values of seeing. In each panel the top pane shows what happens when the noise is under-estimated and the bottom shows what happens to the false positive rate when the noise is over-estimated. In all cases the counts are for the total number of false positives in a single 4000x4072 pixel random Gaussian field.
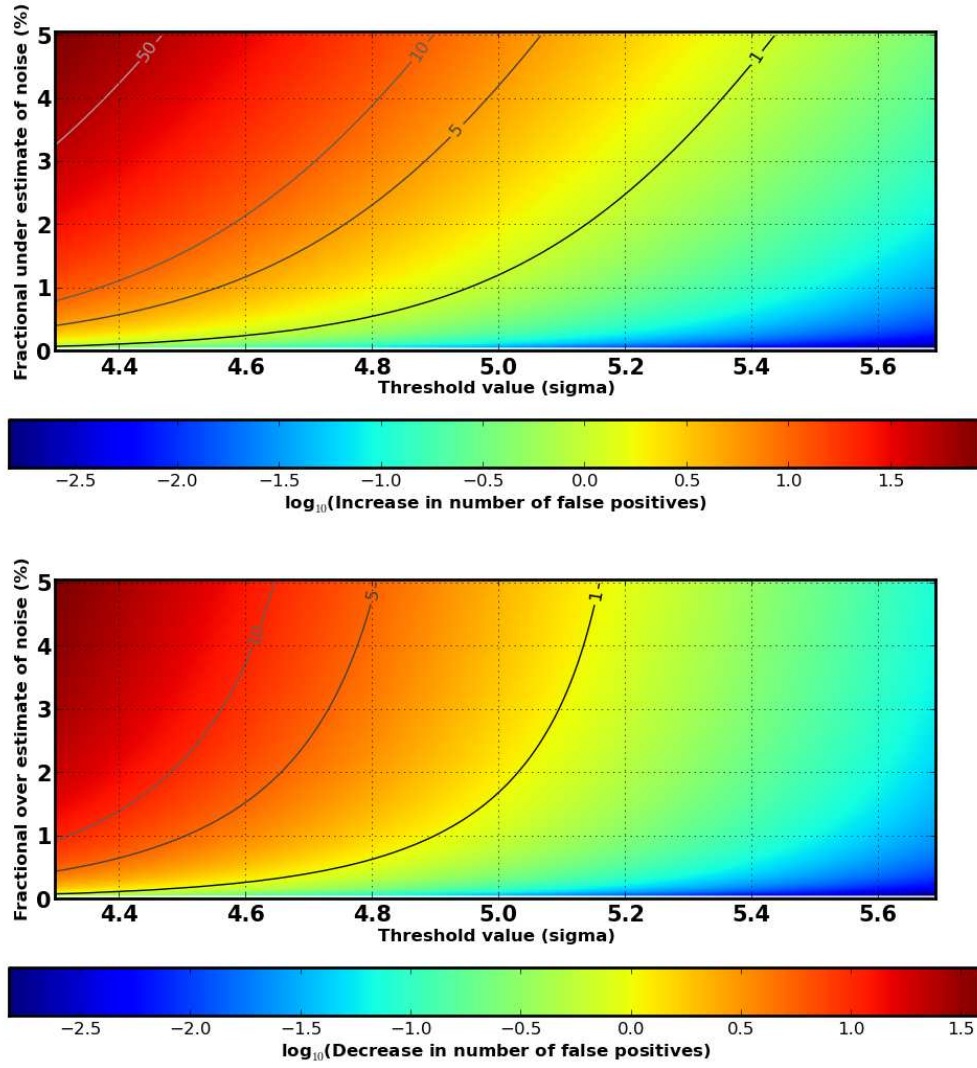
False Positive Rate Dependence on Detection Threshold for seeing = 0.88"
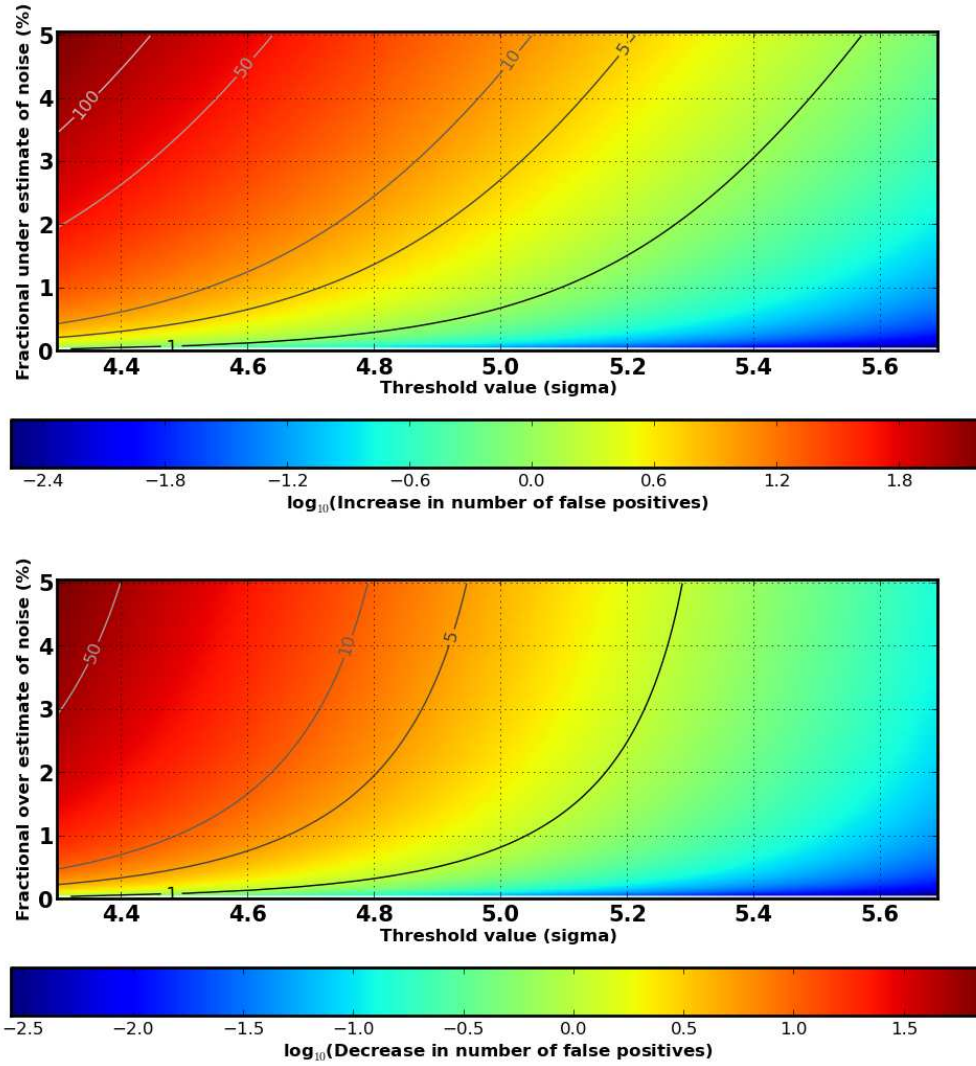
(b) 0.88″ FWHM

Figure 8

False Positive Rate Dependence on Detection Threshold for seeing = 1.20"

(c) 1.2″ FWHM

Figure 8

The signal of a detection may also be misestimated by over or under subtracting the background. The result of errors in modeling the background is to change the ratio of positive to negative false detections. The top pane of Figure 7 shows how the ratio of sources of different polarities changes as a function of cutoff threshold. At a threshold of 5σ one can observe 50% more false detections of one polarity over the other with only a 1% error in background modeling. This, of course, also means that *different* false positives are detected than if the background is subtracted perfectly. Interestingly, the total number of false detections is not nearly as strong a function of background modeling errors. As the number of one polarity decreases, the number of the other polarity increases almost in proportion. This is shown in the bottom panel for Figure 7.

The fact that the total number of false positives depends strongly on the accuracy of the estimation of the variance but not the background estimation, coupled with the fact that the ratio of positive to negative sources should be unaffected by variance misestimation but is very strongly correlated with background modeling, gives us leverage to estimate the overall accuracy of both noise estimation and background modeling with a single measurement of the number of positive and negative orphans.

6.2   Number of KernelCandidates

We effectively decreased the numbers of KernelCandidates in the spatial solution by increasing the size of SpatialCells used to model the solution, while keeping the number of objects used from each cell the same. We increased this configuration from the default size of 128 pixels to (256, 384, 512, 640) pixels. We tracked the numbers of false detections, and the number of used KernelCandidates, and these curves are displayed in Figure 9. The *left* panel shows the numbers of false detections, while on the *right* the numbers of KernelCandidates used, summed across all 9 sensors.

We note that in the pre–filtered data the numbers of false detections are not strongly dependent on the number of constraints on the spatial solution. For visits v6866601 and v6866603 this number is even seen to decrease with decreasing KernelCandidates. This suggests that in pre–filtering there is not a significant difference in quality of fit when using $\sim$600 vs. $\sim$300 KernelCandidates per sensor. The overall fit is demonstrably worse when using only $\sim$100 constraints on the model (cell size = 640). The indication is that 300 constraints on the spatial model (which has 28, 28, and 21 terms per basis for visits v6866601, v6866602, and v6866603 respectively) appears sufficient in the case of pre–filtering.

In the case of post–filtering, aside from the deconvolution case, the number of false detections continually increases as the number of KernelCandidates decreases. This indicates that the best mode to operate in when post–filtering is to maximize the number of constraints on the model.

6.3   Registration Errors

We implemented two simple perturbations of the inputs to the image–to–image RegisterTask: we first added a random offset to each object's (x,y)-coordinates, with an amplitude that was specified in the Config and multiplied by a random number pulled from a normal distribution; and we added a DC offset to the coordinates at an amplitude specified in the Config. These offsets, added to the Source coordinates, will cause misalignments of the objects in the registered images, as
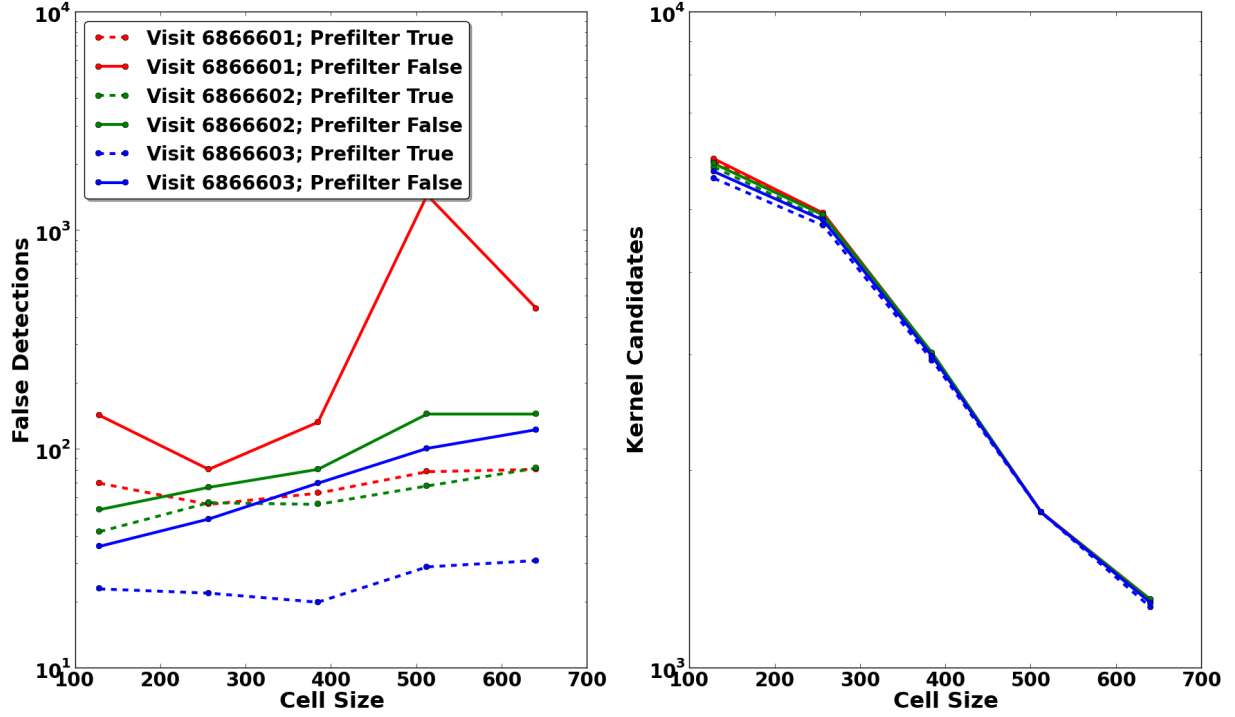
Figure 9: We show the number of false detections (*left*) and `KernelCandidates` used in the spatial kernel fit (*right*) as a function of `SpatialCell` size. Increasing the cell size, while keep all other things equal, effectively decreases the number of constraints on the spatial model. As shown on the left, the pre–filtered data (*dashed* lines) have a less strong dependence on the number of constraints compared to the post–filtered data (*solid* lines). The pre–filtered data show essentially similar quality when using 300 vs. 600 `KernelCandidates`, while in the post–filtered data the number of false positives increases by nearly a factor of 2 (from cellsize 128 to 384).

the registration is done assuming the specified positions are correct. In this way we are able to investigate how random uncertainties and bulk astrometric offsets impact the false positive rate. We explicitly do *not* investigate spatial variation in these offsets, for example using a pincushion distortion. We anticipate that this latter effect will be most important for spatial interpolation and extrapolation of the matching kernel, yielding a dipole residual field associated with the distortion. However, we start with the basics here to investigate the individual terms in such misalignments.

6.3.1   Coordinate RMS

We perturbed the coordinates of each template Source that was input to RegisterTask with ampli-tudes of (0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.3, 0.4, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0) pixels, and to randomize the effects we multiplied each offset by a number pulled from a normal (0,1) distribution. The output RMS reported by RegisterTask was noted to track these offsets.
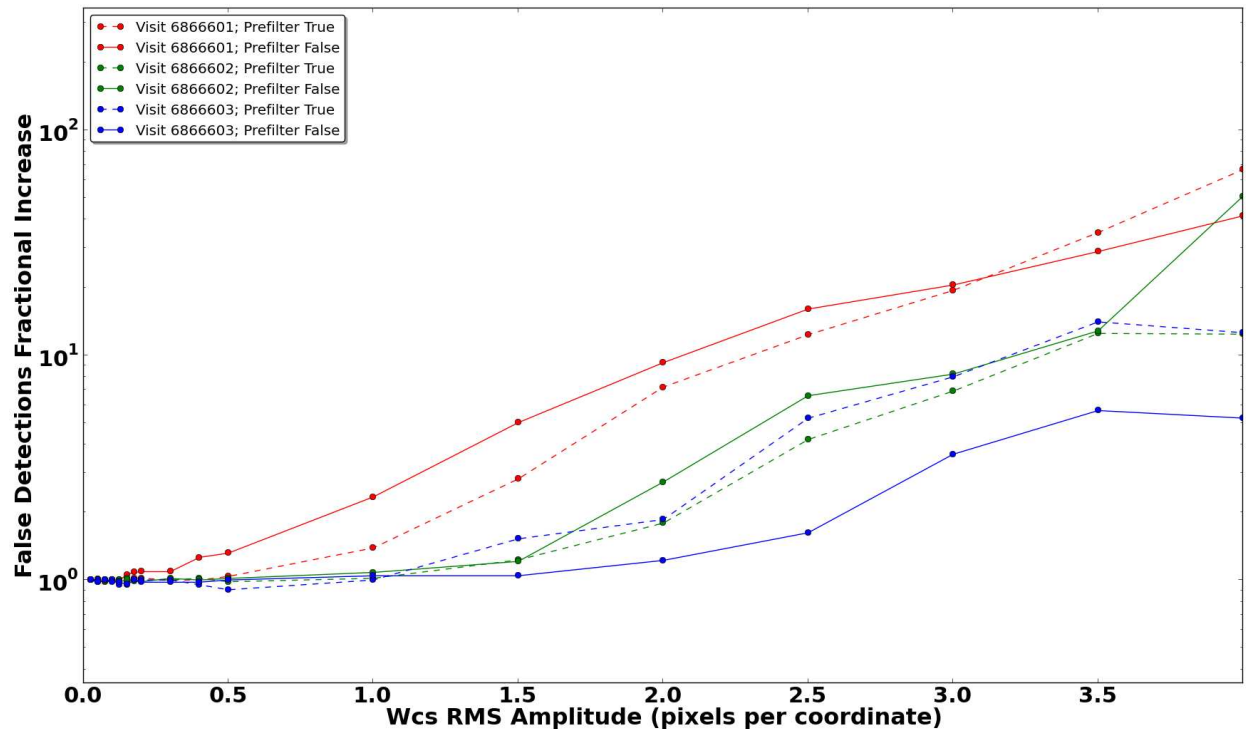

Figure 10: Fractional increase in false detections as a function of the amplitude of random astromet-ric offsets added to the coordinates of sources before image-to-image registration. The *red*, *green*, and *blue* lines correspond to visits v6866601, v6866602, and v6866603 respectively, while the *dashed* and *solid* lines correspond to pre–filtering and post–filtering, respectively.

Figure 10 shows how the numbers of false positives scales with this RMS. Note that the distribution is very flat out to 1.0 pixel (0.2") for all images except for the deconvolution configuration. The number of false positives doubles after an RMS of 1.5–2.0". This test indicates that, for this very simplified case, random astrometric uncertainties do not strongly impact the difference image

quality. This may change with more realistic distributions of stellar brightnesses, if individual objects pull the fit more than others due to inverse variance weighting. In this production, with the stars having the same brightness, no single random offset is likely to drive the fit.

6.3.2 Coordinate Offsets

We offset the coordinates of each template Source that was input to RegisterTask with amplitudes of (0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.3, 0.4, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0) pixels. Post–registration, this will offset the positions of sources in the two images by the desired amount.
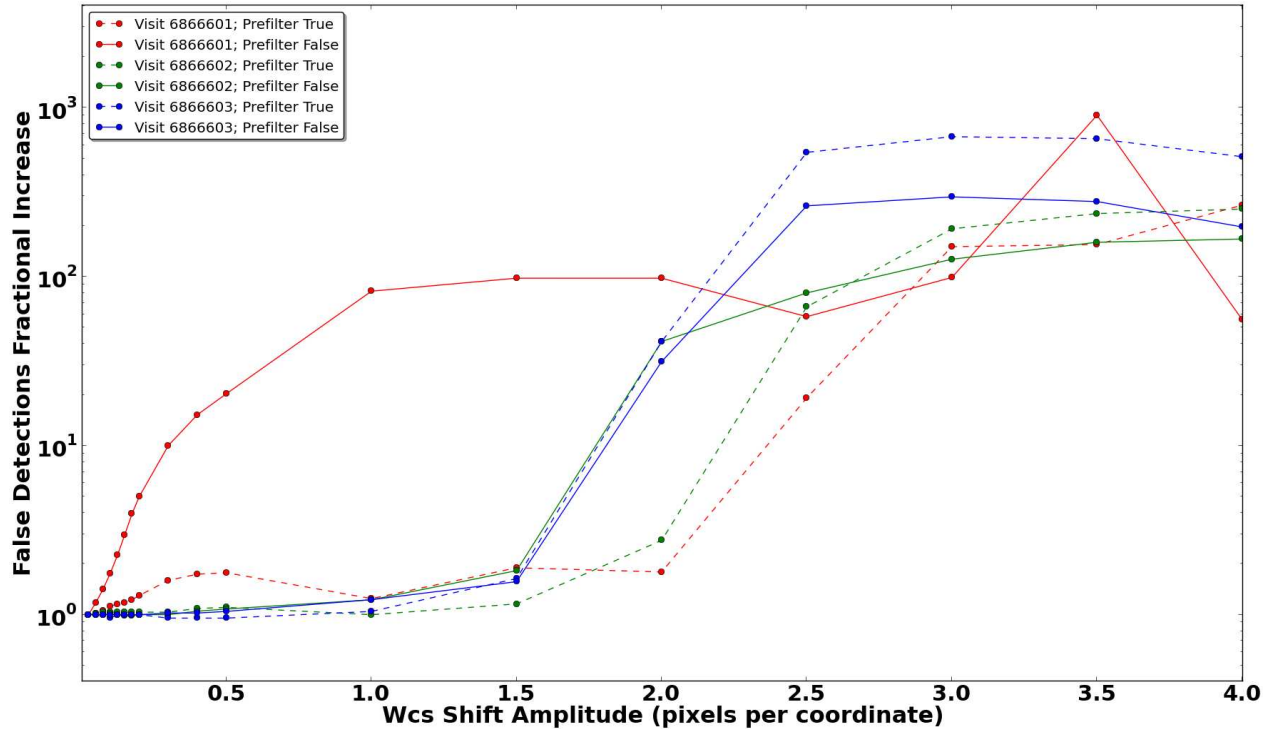

Figure 11: Fractional increase in false detections as a function of bulk astrometric offsets added to the coordinates of sources before image-to-image registration. The *red*, *green*, and *blue* lines correspond to visits v6866601, v6866602, and v6866603 respectively, while the *dashed* and *solid* lines correspond to pre–filtering and post–filtering, respectively.

Figure 11 shows how the numbers of false positives scales with this shift. We note that the distribution is very flat out to nearly 1.0 pixels (0.2") for all images except for the deconvolution configuration. The number of false positives doubles after a shift of 1.5". This test indicates that, for this very simplified case, bulk astrometric offsets of the degree expected from nightly processing should not drive the false positive rate.

Figure 12 demonstrates the ability of the PSF–matching kernel to redistribute flux to account for bulk astrometric misalignments. Since these offsets are common–mode amongst all KernelCandidates, the basis shapes themselves drive the pipeline's ability to compensate for them. Having a Gaussian
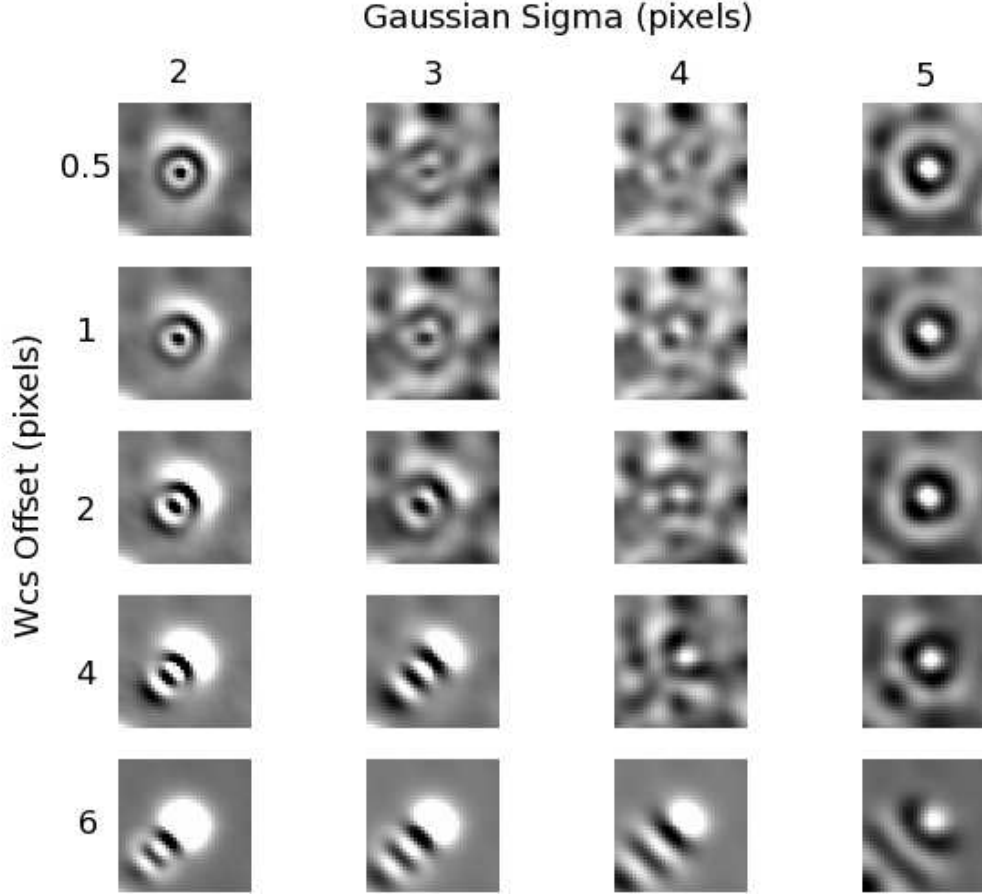
Figure 12: Figure illustrating the ability of the kernel basis to compensate, locally, for the effects of misregistration. Each element of the grid shows a single `KernelCandidate` local difference image (visit=6866603, doPreFilter=True). Columns indicate the Gaussian sigma of the basis, which is designed to have one Gaussian with the prescribed width, modified to fourth order. Rows indicate the amplitude of the astrometric offset that was introduced into the Source positions before image-to-image registration. In the *top* row, we see that even with a small astrometric offset, the sigma=2 basis is unable to produce a quality subtraction, because the width is inappropriate for matching the two input PSFs (the theoretical Gaussian matching sigma here is 3.4 pixels). At sigma=4 the basis is able to produce a quality subtraction, and by sigma=5 the basis is too large to match the PSFs. In *row 4*, we show that even for an astrometric offset of 4 pixels (0.8"), the sigma=4 basis can produce a reasonable difference image without structured residuals. However, by *row 5*, with an offset of 6 pixels, none of the Gaussians provide a successful local difference image. The ability of the basis to compensate for bulk astrometric offsets is a function of the kernel width, and the kernel width is itself a function of the two input PSF FWHMs. Thus there is a complicated dependence of the ability of the kernel to shift the centroids of stars. Roughly, in our implementation, the basis may compensate for local astrometric residuals at a scale up to $\beta \times \sqrt{\sigma_S^2 - \sigma_T^2}$, which is the width of the largest Gaussian in our bases.

Variance Ratios: Empirical / Propagated			
Visit	$D_{pre} = D'_{pre}$	D_{post}	D'_{post}
v6866601	1.012 ± 0.002	1.018 ± 0.005	0.599 ± 0.017
v6866602	1.010 ± 0.002	1.007 ± 0.001	1.046 ± 0.002
v6866603	1.015 ± 0.003	1.008 ± 0.001	1.045 ± 0.003

Table 4: Ratio of the empirical variance in the difference images, calculated via (0.741 times the interquartile range)^2, to the median of the variance plane. In all cases (except the deconvolution configuration) the variance plane represents an *underestimate* of the true variance in the images. We report the mean and RMS across all sensors for the best configurations.

basis function whose width is commensurate with the amplitude of the offset enables the software to place significant power in this basis term. We expect that, if these misalignments are a vector field, the ability of the spatial kernel model to account for the spatial variation of the misregistration will begin to drive the solution. The take–away from this analysis is that the PSF–matching basis appears flexible enough to account for, individually, random scatter and bulk offsets in the WCS solution. This narrows the focus down to understanding how spatial variation in the WCS–based registration may drive the difference image quality, and what requirements need to be put both on the kernel basis and its spatial variation.

6.4   Variance Misestimation

Because these images will have gone through multiple convolutions before reaching the detection stage, it is expected that the propagated per-pixel variance will not provide an exact representation of the empirical variance in the image planes (Price & Magnier, 2010). Because the astrometric resampling uses a warping kernel that is designed to preserve the noise properties of the images, we expect that the kernel convolution and PSF–filtering will provide the largest sources of such error.

To quantify this, we first compared the empirical variance in the image planes with the level tracked by the variance planes, for images D and D'. We calculated the ratio of the interquartile range of unmasked pixels in the image plane, multiplied by 0.741 and then squared, with the median of unmasked pixels from the variance plane. The former represents the empirical variance, while the latter the propagated variance. We investigate their ratio (empirical/propagated) at 2 stages in the processing: after creation of the difference image (image D), and immediately preceding detection (image D'). Recall that for pre–filtering, $D = D'$. These results are summarized in Table 4. For pre–filtering, we find that the variance plane typically underestimates the true variance by 1% for all visits. When using post–filtering, the deconvolution visit v6866601 yields a similar underestimate in the difference image, but for the other two visits the variance is tracked to better than 1%, with a small RMS. However, when post–filtering with the PSF, the variance becomes *overestimated* by nearly a factor of two for deconvolution visit v6866601, and *underestimated* by $4-5\%$ for the other visits.

We next investigated the variance properties of the images that are input to the detection stage, i.e. the difference image itself in the case of pre–filtering, and the difference image convolved with its PSF in post–filtering. These results are summarized in Table 5. We find that the image planes

Variance Ratios: D'_{post}/D'_{pre}		
Visit	Empirical	Propagated
v6866601	1.011 ± 0.016	1.710 ± 0.049
v6866602	1.001 ± 0.001	0.967 ± 0.001
v6866603	1.002 ± 0.001	0.974 ± 0.002

Table 5: Ratios of the variance planes immediately before detection D'. Ratios are in the sense of the variance of the post–filtered image divided by the variance of the pre–filtered image. The first column represents the ratios of the empirical variance, calculated via (0.741 times the interquartile range)^2. The second column reports the ratios of the medians of the propagated variance planes. We report the mean and RMS across all sensors for the best configurations.

have very similar empirical noise properties at the detection stage, with the deconvolution visit having larger variance by 1% but the other visits having essentially equal noise properties, with a bias towards the post–filtered data having slightly higher variance (while the RMS is small, all values are greater than 1.0)[‡]. The medians of the variance planes tell a different story though. For the deconvolution visit, the median per–pixel variance is 70% higher. While this will suppress the detections of false positives, it will also significantly suppress the detection efficiency for true variability. For the other visits, the post–filtered per–pixel variance is relatively *lower* by $3-4\%$. This will increase the sensitivity of the post–filtered data to false positives; as shown above, the rate is indeed larger using the post–filtering pipeline. This analysis demonstrates that this is primarily due to an *underestimate* of the per–pixel variance in the post–filter processing: while the pre–filtered variance is a 1% underestimate, the post–filtered variance is a 5% underestimate. As outlined by Price & Magnier (2010), this due, at least in part, to loss of pixel covariance when undergoing multiple convolutions, as covariance is not currently tracked in the DM stack. In pre–filtering, the science image undergoes a single convolution, and the template image a single convolution. In post–filtering, the template image undergoes a convolution, is subtracted from the science image, and then undergoes a second convolution with the Psf. As we have demonstrated here, this second convolution is where our variance bookkeeping lags the empirical variance.

6.5   Detection Threshold

Obviously the number of false positives is strongly dependent on the detection threshold. In order to check the scaling of our false positive rates with detection threshold, we ran the same config (see Table 2 for the config parameters) at several detection threshold values. As discussed in Section 5.1 one can predict the number of false positives expected in a sensor given seeing and threshold values. Values of the number of false positives (both +ve and -ve) expected in LSST size detectors are given in Table 3.

As discussed in the Section 6.1, the accuracy of the noise estimation in the detection is crucial to understanding the false positive rate. In the previous section we quantify the under–estimation of the variance by the variance plane in the case of pre– and post–filtering. In our analysis of the

---

[‡]For completeness, we note the median differences between the image planes at the time of detection are 3–4 counts (or 0.2 sigma relative to the empirical variance) for v6866601, 0.2 counts (0.02 sigma) for v6866602, and less than 0.1 counts (0.01 sigma) for v6866603.

threshold–dependence of our false detection rate, we correct for this under–estimation by multiplying the requested detection threshold by a scaling factor in order to put the counts from each algorithm on the same normalization (Table 4).
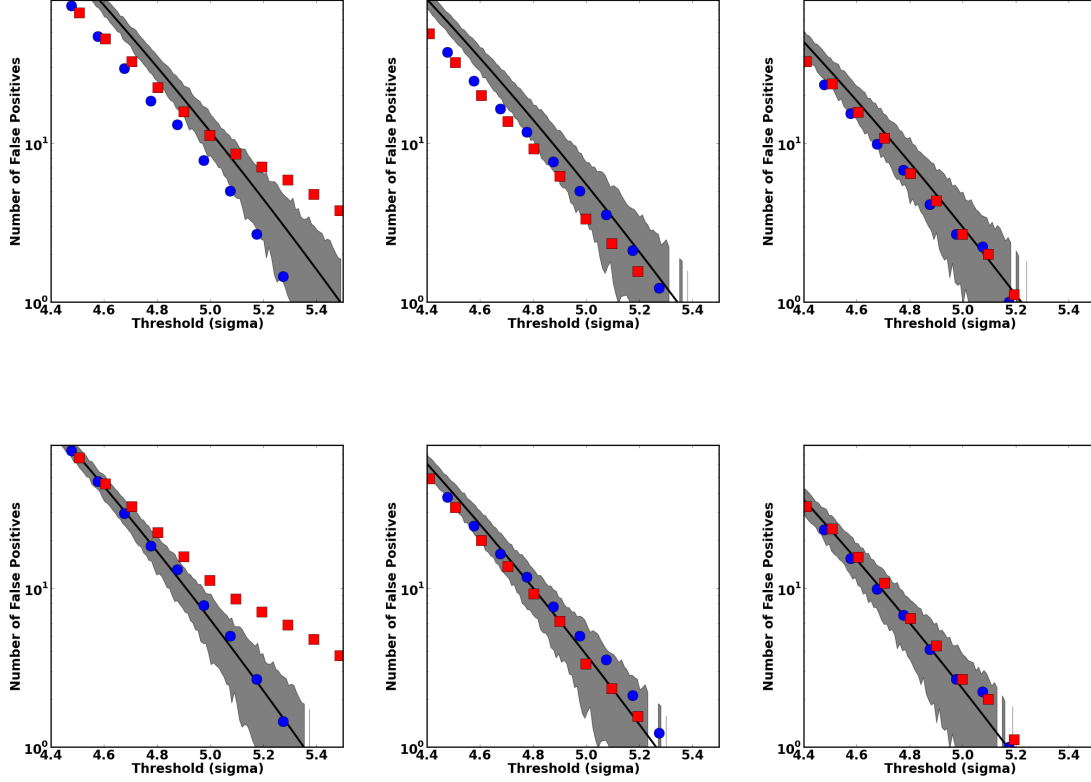

Figure 13: Here we show the comparison of detected false positive count to the analytic prediction. In all cases the blue circles are the pre–filter case and the red squares are the post–filter case. The top row has had the counts corrected for the underestimate of the noise by the variance plane (Table 4). In the bottom row we undertaken a 1–parameter correction to align the predicted and measured values. See Section 6.5 for how this correction is applied. The 2.5%, 1.5% and 1% scaling in k correspond to a 5%, 3%, 2% over-estimation of the image variance, respectively from Left to Right in the bottom pane. The shaded area is the 1-σ confidence envelope determined from Monte Carlo simulations (Section 5.1).

Figure 13 shows the results of this analysis. In all figures red squares are post–filtered data and blue circles are for the pre–filtering technique. All values are for the number of false positives in a 4000×4072 sensor. Using a Monte Carlo approach we estimate the variance on the predicted false positive rate as a function of threshold. The 1-σ confidence envelope is shown as shaded grey about the model line in dark black. From left to right the panels are for v6866601 (0.6″ FWHM), v6866602 (0.88″ FWHM) and v6866603 (1.2″ FWHM) respectively. Except in the case of deconvolution in the post–filtering case (v6866601) the count rates between the two algorithms agree very well when corrected for the observed under–estimation of the noise by the variance plane. In the case where

post–filtering requires a deconvolution, the high S/N detections come primarily from detection of the ringing around stars.

In all cases the empirical results are below the predicted results indicating that the number counts are smaller than the theoretical floor. This is not entirely unexpected, as each curve is strongly correlated, being a cumulative distribution, and represents only 9 samplings of the background random Gaussian field. We find that we can perform a 1–parameter correction to Equation 2 curves that yields both the same slope and zeropoints as the theoretical curves, with the exception of the deconvolution visit.

Specifically, we define scaling factor k to correct the counts at a given threshold:

$$n_{corrected}(> \nu) = n(> k\nu) \tag{4}$$

For our data we scale ν in the model by $k = 0.975, 0.985$ and 0.99 for v6866601, v6866602, and v6866603 respectively. The data plotted with these corrections are shown in the bottom row of Figure 13. These results suggest our variance estimates are too *high*, and we are therefore detecting too *few* false detections. This does not agree entirely with the analysis of the empirical variance, Section 6.4 and Table 4, which suggest that our variance planes *underestimate* the true variance. However, this makes clear that while we are at the level of the theoretical floor in false detection rate, it is imperative to track the variance in our data at the 1% level or better to optimize both the false detection rate, and the true detection efficiency. This will be a subject of the Summer 2013 work.

7   Results: Correlation with KernelCandidate Statistics

In Figures 14 and 15 we correlate statistics calculated on the control sample of `KernelCandidates` with the numbers of false detections that come out of detection and measurement, for the pre–filtering and post–filtering cases respectively. We do this for statistics that represent the median standard deviation of the pixels in the `KernelCandidate` SPATIAL difference image (KCDiffimStDev), the median reduced chi2 of the pixels in the difference image (KCDiffimChiSq), and median MSE of the residuals (KCDiffimMseResids). These values are defined in Section 3.3. There is one point in each figure from each sensor in each of the 72 production runs per pre–filter and post–filter configuration.

For post–filtering, Figure 14 indicates that these metrics correlate strongly with each other, meaning there is likely little advantage to looking at more than one metric. The deconvolution visit in *red* is a clear outlier in this distribution. While there is a relationship between the values of these metrics and the numbers of false positives, it is relatively flat until $\sim 10^2$ false detections per sensor; the scatter below this threshold is similar to the trend, meaning there is no single metric studied that may be used to predict the numbers of false detections during the fitting process, within the range of interest (< 10 false detections per sensor).

For pre–filtering, Figure 15, the slope between 10^0 and 10^1 false detections is steeper, but still commensurate with the scatter within that range. The metrics are still strongly correlated with each other, but not to the degree as for the post–filtered statistics. A less–noisy relationship might

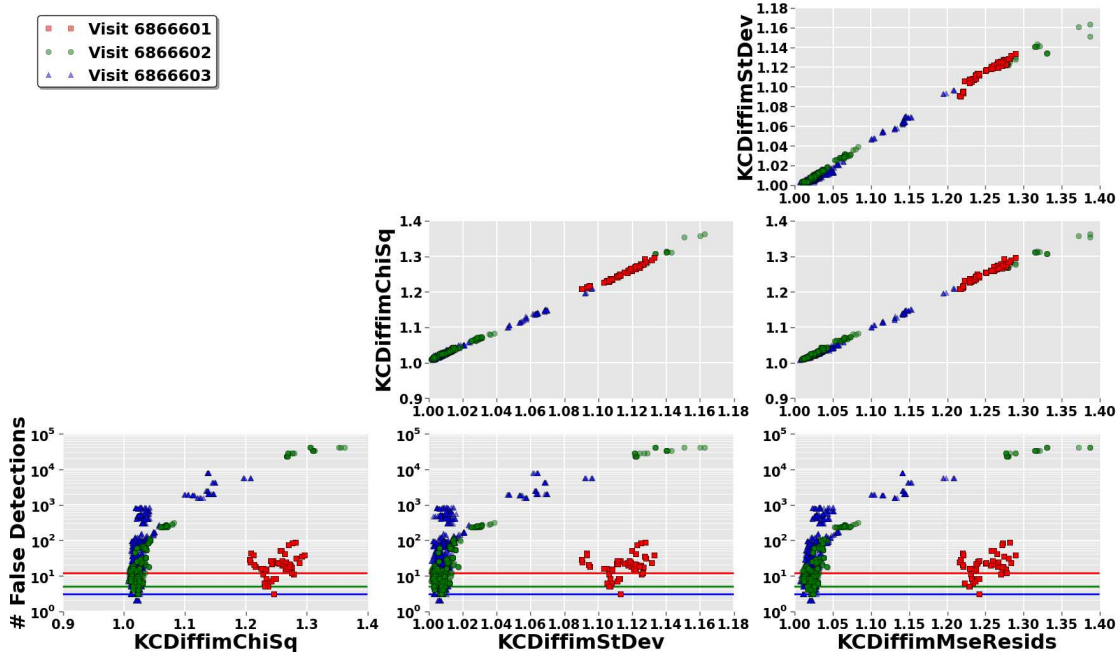be obtained through a linear combination of these metrics, but we have not fully investigated this option.

Figure 14: Correlation of the statistics calculated on the control sample `KernelCandidates` during the model fitting with the number of false positives per sensor coming from detection and measurement, for the **post–filtering** pipeline. Results for visit v6866601 are shown as *red squares*, for v6866602 as *green circles*, and for v6866603 *blue triangles*. Three of the statistics, described in Section 3.3, are presented. They clearly correlate with each other. For the deconvolution visit v6866601, the statistics are very poor even for the "best" solutions. Points come from all 144 production reductions of the same data, and detection at 5–sigma. The minimum numbers of 5–sigma false detections per–sensor (Table 3) are given by the horizontal lines.

We do note that these distributions are internally correlated, since they come from many runs of the pipeline (in different configurations) on the same data. A better method to use to investigate this is individual pipeline runs on much larger amounts of data, where the different points are uncorrelated. We plan to quantify these correlations in further detail as we continue our work in Summer 2013.

Using this simplified data set, we can begin to look at how we may predict optimal configs based on measured statistics. In particular, we investigate the MSE, and show how it varies as a function of config in Figures 16 and 17. We find that the MSE of the fitted sample (Figure 16) continually decreases as the complexity of the model increases. However, for the control sample (Figure 17), a minimum is achieved in all cases. This minimum value suggests the point where the model starts overfitting, and is itself a method to suggest the optimal configuration for image differencing, along with false positive rate. However, MSE of a control sample is a metric that may be applied to data where the truth is not known, and thus has potential to be used in the nightly processing. For more
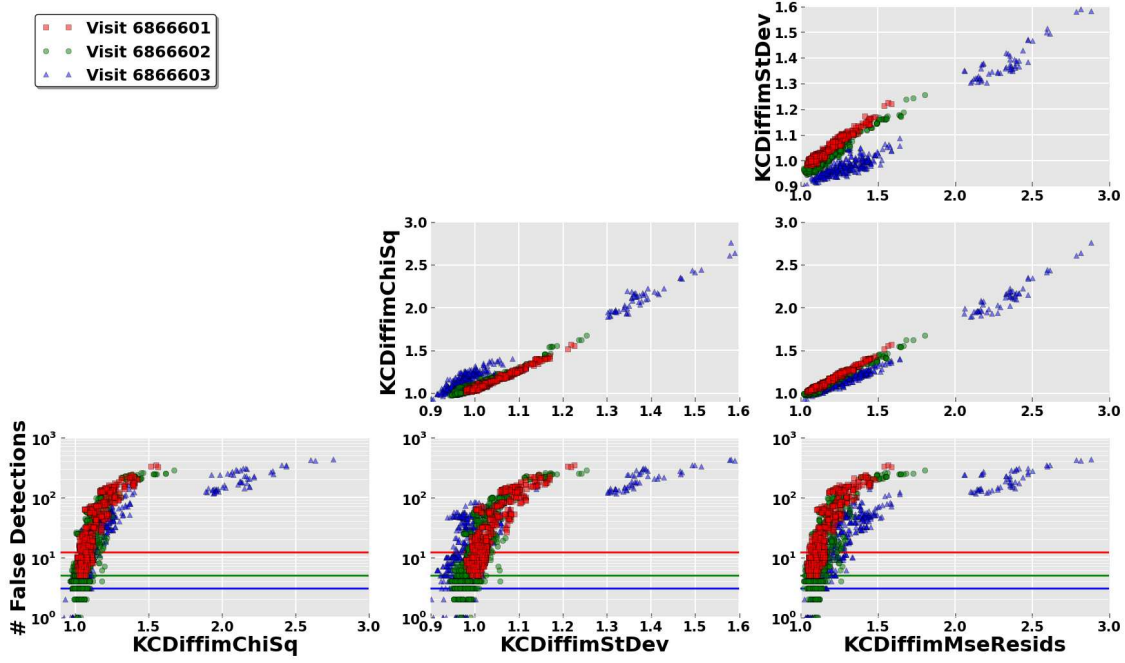
Figure 15: Correlation of the statistics calculated on the control sample `KernelCandidates` during the model fitting with the number of false positives per sensor coming from detection and measurement, for the **pre–filtering** pipeline. Results for visit v6866601 are shown as *red squares*, for v6866602 as *green circles*, and for v6866603 *blue triangles*. Three of the statistics, described in Section 3.3, are presented. They clearly correlate with each other, and also tend to correlate with the number of false detections (bottom row). The minimum numbers of 5–sigma false detections per–sensor (Table 3) are given by the horizontal lines.

(a) PREFILTERED   (b) POSTFILTERED
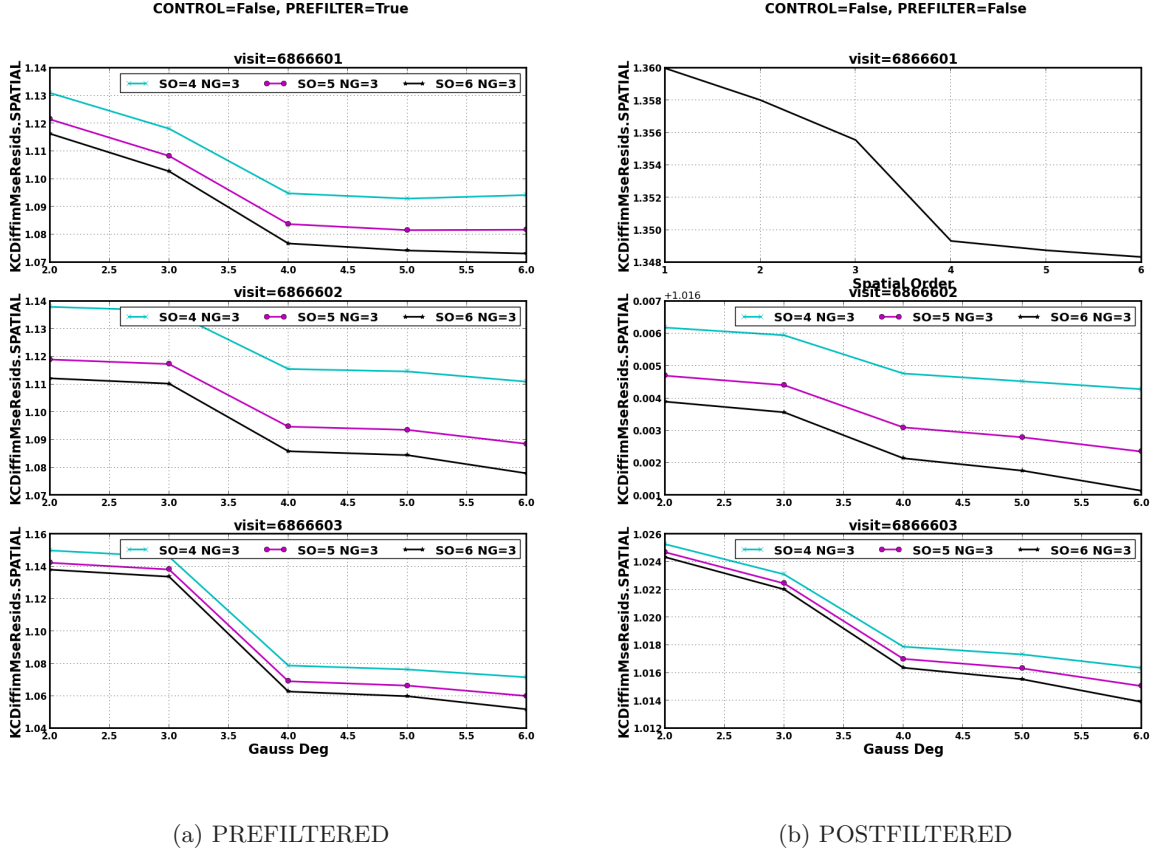
Figure 16: We plot the $\langle MSE \rangle$ for each visit using the sample used in the model fitting. The left is for the pre–filtering case and the right is for the post–filtering case. In each case we have plotted only the nGauss=3 and spatial order=(4,5,6) because we find that a spatial order of at least 4 is needed. The top pane in the post–filtering case just shows the spatial order since the other values are held constant in the case that deconvolution is requested. Notice that there is typically no minimum.
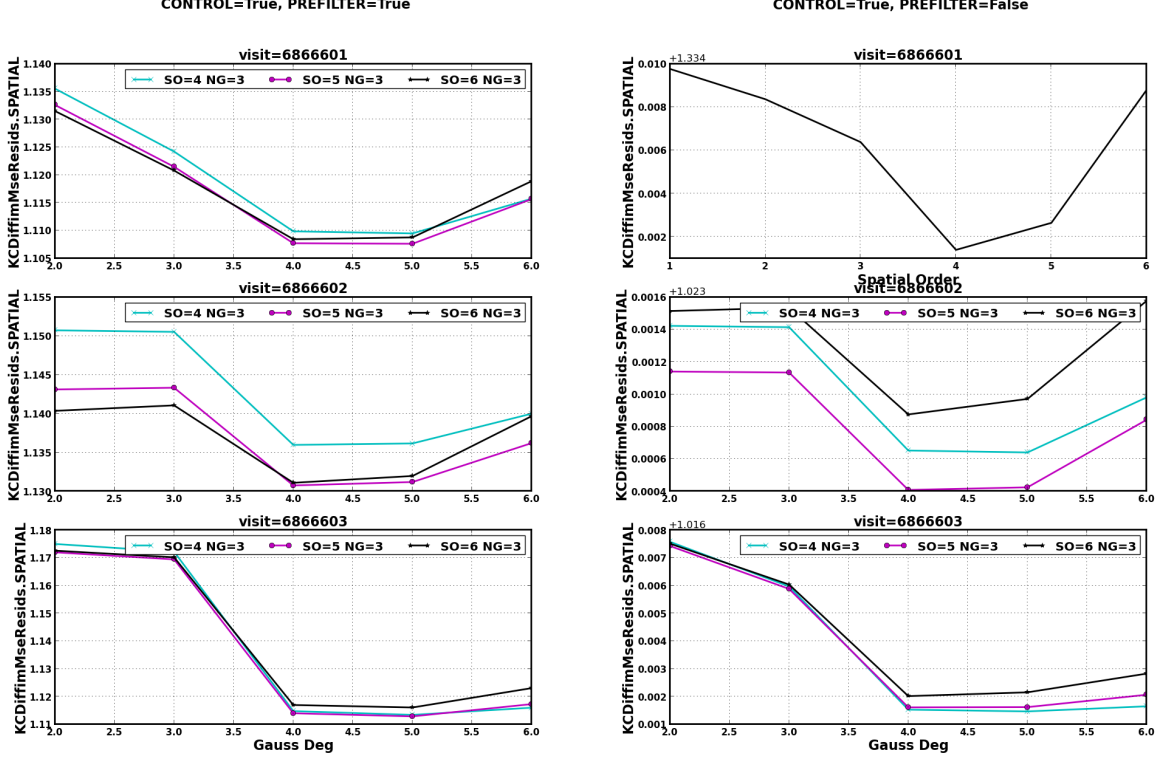
(a) PREFILTERED

(b) POSTFILTERED

Figure 17: We plot the $\langle MSE \rangle$ for each visit using the control sample which was not used in the model fitting. The left is for the pre–filtering case and the right is for the post–filtering case. In each case we have plotted only the nGauss=3 and spatial order=(4,5,6) because the minimum MSE came from these for all visits. The top pane in the post–filtering case just shows the spatial order since the other values are held constant in the case that deconvolution is requested. Notice that all visits have a minimum. Compare this to the same statistic calculated on the fitted sample in Figure 16 where there is no minimum.

Best Configs: From Minimum MSE				
Visit	Pre–filter	nGauss	degGauss	spatialOrder
v6866601	True	3	4	5
	False	3	1	6
v6866602	True	3	4	6
	False	1	3	5
v6866603	True	3	5	5
	False	3	5	5

Table 6: The best configs as derived from the minimum in the measured MSE.

discussion on this see Section 8.11 in "The Second Best Book in the World" (i.e. Ivezić et al., in press).

We note that the config at the minimum MSE value is not always the same config that yields the lowest numbers of false detections. Table 6 lists the configs at the minimum of the MSE for the control sample (compare to Table 2). We have also looked at per–sensor MSE values and per–sensor false positive rates. In both cases, the config that produces the minimum number of false positives is variable from sensor to sensor. It is worth investigating how much the false positive rate goes down if per–sensor optimal configs are used instead of per–visit optimal configs. On the other hand, because the false positive rate is relatively flat beyond a critical configuration (Figure 5), we are not strongly sensitive to getting the per–sensor tunings perfect. When we look at the MSE per chip and compare with the minimum false positive config we find good agreement, in general. The findings are summarized in Table 7. Because the MSE is sensitive to over fitting, we observe that it tends to lower complexity configs even if the false positive rate is slightly higher. We will continue to investigate this metric in Summer 2013.

8   Results: Measurement

Figure 18 provides a mosaic of absolute–value S/N measurements for the `DiaSources` detected at 5–sigma. The *top* panel shows the PSF flux measurements reported in the post–filtered analysis, while the *bottom* panel shows the filtered flux measurements in the pre–filtered analysis. Both distributions are strongly peaked at 5–sigma, indicating we are not grossly mismeasuring fluxes or flux uncertainties in either case. A detailed estimate of the accuracy of the pre–filtering photometry will be undertaken in the summer 2013 work.

9   Results: Computational Performance

For this production, the understanding of the algorithm and optimization of the false detection rates took priority over optimizing the computational performance of the algorithms. In particular, debugging hooks were implemented and workarounds of problems were devised with the focus on understanding the core PSF–matching algorithm. Therefore there are several key ways in which the production as–implemented will differ from how the code is used in Nightly and Data Release

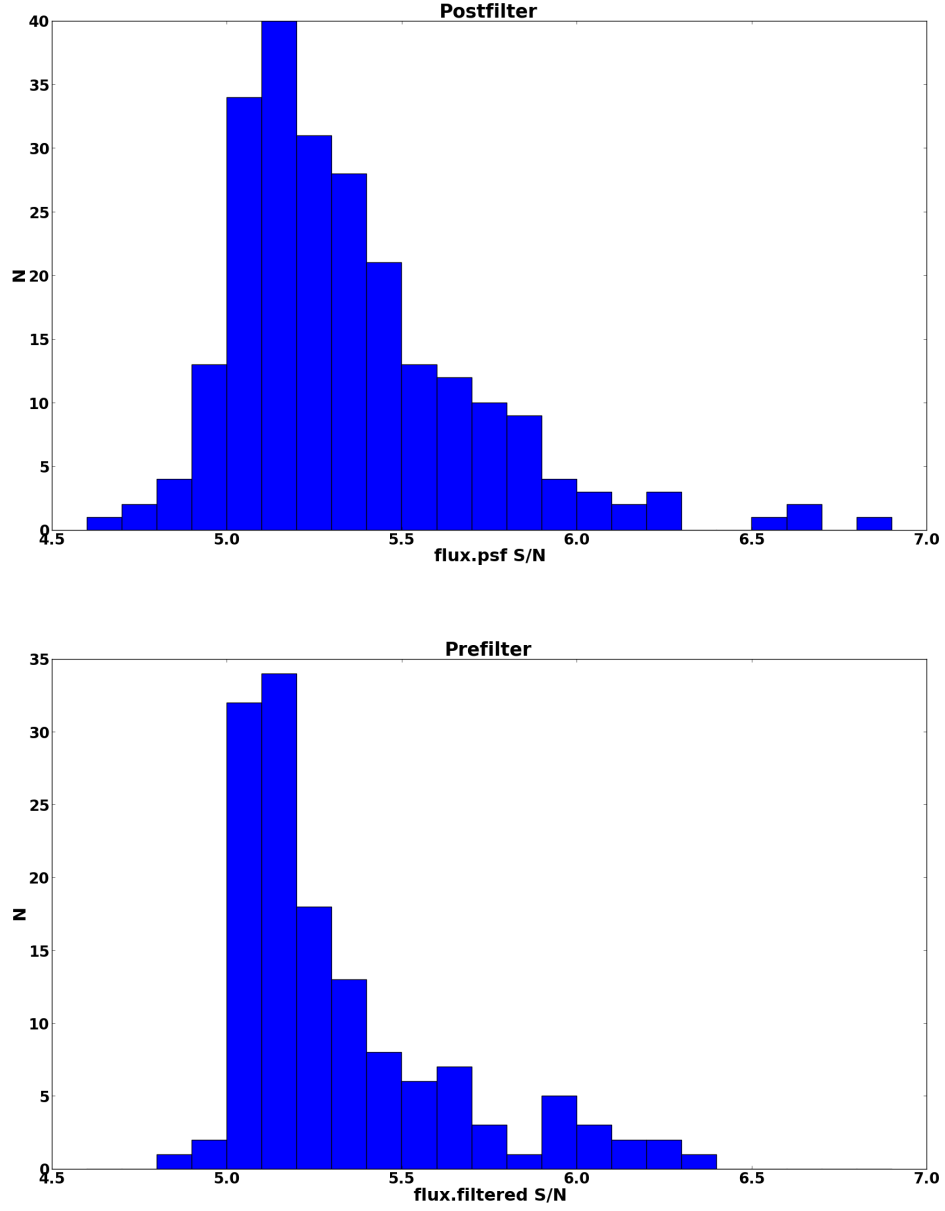Figure 18: Histograms of measured | signal–to–noise | for sources detected at 5–sigma in the best post–filtered (*top*) and pre–filtered (*bottom*) data. In post–filtering, we use the PSF flux, while in pre–filtering we use the filtered flux measurement.

Processing. We outline these differences below.

The first difference is that due to problems with the `meas_astrom` package, detailed in Section 11.1, we were unable to exercise the envisioned data flow where we query a coadd repository for the template image. Instead we used a single deep `calexp` as the template image. The astrometric registration of these images was trivial, as the images almost exactly overlap and are exactly the same size. Thus the timings of the `imageDifference:register` task are *underestimates* of the expected astrometric registration subtask computational performance.

Second, the core performance of the `imageDifference:subtract` task will be significantly impacted by debugging hooks that were put in place specifically to generate the metrics being evaluated in this production. The core functionality, `matchMaskedImages`, does the convolution kernel fitting and actual convolution of the template image, and will be least impacted by these add–ons. The reported timings of `matchMaskedImages` are thus an *overestimate* of the method's performance.

The reported `imageDifference:detection` task performance is an accurate representation of its computational requirements. However, the `imageDifference:dipolemeasurement` task represents a significant overestimate of the true dipole measurement requirements. In particular, the PSF–flux dipole measurement is explicitly sub–optimal, in that its centroid fitting algorithm is a 4–deep `for` loop.

10 Conclusions

- At the 5–sigma detection threshold, for the range of seeings considered in this production, the theoretical rate of false detections per 4k x 4k sensor numbers 3–12, depending on seeing. **We are able to realize this theoretical floor**. This function is steep and seeing dependent; by increasing the detection threshold to 5.5 sigma this number may be lowered to less than 1 per sensor down to seeing of 0.6" (Table 3). This translates to of order $2 * 10^5$ per night, assuming 1000 visits / night. The steepness of the false positive function will enable control of the number of false alerts by small modifications of the detection threshold.

- Due to the steepness of this function, the number of false detections is strongly dependent on the variance being correct, at the 5–sigma detection threshold. When detecting at 5–sigma, having the noise underestimated by 2% leads to an increase in false detections by a factor of 1.5–2 (Figure 8).

- When strongly deconvolving, the post–filtered variance is 70% *higher* at the time of detection compared to the pre–filtered data (Table 5). This is consistent with the understanding that deconvolution increases high frequency noise in the images, and provides a *lower* detection efficiency in deconvolved data. Pre–filtering is clearly preferred to post–filtering in the case of significant deconvolution (v6866601).

- For non–deconvolved image subtraction, pre–filtering performs as well as post–filtering in terms of the number of false positives, when corrected for the variance underestimate (Figure 13).

- An empirical computation of the variance from the image plane, compared to the median propagated variance in the variance plane, indicates that we consistently underestimate the

variance in pre–filtering by $1 - 2\%$, and in post–filtering by $4 - 5\%$ when not strongly de-convolving (Table 4). The post–filtering issue is shown to arise from the PSF–filtering of the difference image immediately before detection. This is due, at least in part, to the double–convolution that is happening to the template image, and our lack of covariance tracking. The dependence on template noise and the number of convolutions will impact the requirements on the variance within the template images (and thereby the number of images going into a template image).

- The empirical variance of the detection images in pre–filtering and post–filtering are similar to within 1% (Table 5). However, the propagated variance in the post–filtered data is $3 - 4\%$ *lower* in the post–filtered data, leading to a larger number of false positives when using the variance plane as the definition of sigma (Table 1).

- The post–filter pipeline can produce difference images with a minor deconvolution (v6866602) at a quality commensurate with a subtraction that uses a smoothing convolution (v6866603).

- We find that the overall number of false detections is not strongly sensitive to bulk background misestimation (Figure 7);

- However, the ratio of positive to negative false detections is strongly dependent on background fitting, with a 1% error in the background causing 50% more false detections of one polarity over another (Figure 7).

- Our empirical negative to positive false detection ratio is 2–3 for pre–filtering, indicating a bias in the background levels of 1–2%. Similar ratios are seen in post–filtering, although they vary from over– to under–subtraction of the background (Table 1). This will be a subject of investigation in Summer 2013.

- The measured false detection rate at 5–sigma is within the variance of the expected false detection rate arising from a random Gaussian field (Figure 13).

- The measured false detection rate is seen to scale with detection threshold in a manner consistent with theory (Figure 13). A 1–parameter correction to the variance by 1–2.5% brings these relationships into almost exact correspondence. Optimization of the false detection vs. true source detection efficiency will require an understanding of the variance at the 1% level.

- The slope of the false detection rate is shallow beyond spatialKernelOrder > 3 and degGauss > 3 (Figure 5).

- The kernel is able to absorb, individually, random scatter in the astrometry (Figure 10) and offsets in the astrometry (Figure 11) of order 0.2" without a significant increase in the number of false positives. Because the pre–filtering Gaussians tend to have larger widths, they are able to compensate for larger offsets.

- The measurement of `DiaSources` detected at 5–sigma are shown to have a commensurate measured signal–to–noise (Figure 18).

- Higher order measurements on filtered sources are not yet available, meaning feature–based classification of systematic false positives is not yet possible.

41

- Several statistics that we are able to calculate at the time of image subtraction are shown to correlate strongly with the numbers of false detections (Figure 14 and Figure 15). These relationships have a shallow slope in the area of interest (number of false detections per sensor less than 10), although there appears to be more discriminatory power in the pre–filtered data.

- Using the mean squared error of a control sample of `KernelCandidates` appears to be a promising way to understand the degree of over/underfitting of the kernel model (Figure 17). This may be a feasible technique to optimize the pipeline on a per–sensor basis.

11   Outstanding Issues

We ran into some issues during this production that required work–arounds. In some cases we did not have the time to drill all the way down to the root cause. We itemize these issues here so that they may be addressed as part of the Summer 2013 work.

11.1   Astrometric Registration

The original plan for this production was to use the single deep exposures (v6866600) as inputs to the coadd process, to exercise use of the coadd process and coadd repository. As such we warped the exposures to coadd patches for use in the image difference task. The register task was unable to solve a differential WCS between the coadd and science `calexp` because in coadds, the parent WCS is for the entire `tract`. Each coadd `patch` thus has a CRPIX at the center of the `tract`, which is typically far off the `patch` image. The TAN–SIP model is unable to perform adequately in this circumstance, as it expands its polynomials around the CRPIX. To get around this we tried re–fitting the coadd patches using a TAN–SIP solution that was solved from scratch. This yielded residuals in the 40–100 mas range, compared to the 100–300 mas previously. Bugs in meas_astrom that were uncovered as a result of this investigation, required reverting to using the deep `calexp` as the template directly; the direct deep `calexp` to science `calexp` registration typically yielded residuals in the range of 4–5 mas.

We investigated potential causes of the large astrometry residuals. To do this, we looked at the residuals for template–to–reference catalog solutions, science image–to–refcat solutions, and direct template to science image registration to attempt to isolate where the astrometry was being degraded. The results are shown in Table 8. We show the results for the nine sensors in the best seeing visit. The solutions for the three steps that do not involve any warping show results consistent with the accuracy we expect to get. We also see the expected trend that the deep exposure solutions have smaller RMS than the solutions for the science images, due to their higher S/N. The coadd–to–science registration depends critically on the WCS fitting and warping process: the WCS of the deep `calexp` is warped into the coadd `patch`; the coadd `patch` is then warped into the science `calexp`'s WCS frame. This places two dependencies on the warping code, and two on the WCS fitting, where this degradation may have occurred. This 1–2 order of magnitude degradation in the WCS solutions shown in Table 8 indicate that astrometry and warping are areas that need to be actively addressed.

11.2 Detection and Measurement on Negative Sources

We see more negative false detections than positive false detections. This drove an investigation into the symmetry of detection and measurement on positive/negative sources. To do this we looked at the results of the image difference task and compared it to the same runs when the difference image was inverted (multiplied by -1) before measurement. The results are as follows:

- Centroiding on negative sources fails almost all the time: 99.7% sources with negative flux have the `badcentroid` flag set and have integer value centroids.

- Whether detected on the original difference image or the inverted image, the `DiaSource` lists are the same length in every case. That is to say `len(afwTable.SourceCatalog.readFits(`*path*`))` returns the same value in both cases. We take this to mean that the same objects trigger detection in both situations although this cannot be directly verified since the objects with failed centroids are not the same in both lists.

- In some instances psfFlux is stored as 0.0 (not `NaN`) without any centroid or photometry flag set in the `Source` object. This happens more than we would expect if it was just a chance event. This could be an uncaught failure mode.

- psfFlux can be measured as NaN with valid centroid values and no other flags set.

- Measured psfFlux can have the same sign in both the initial and inverted image in 0.1% of measurements.

- Some sources have the `EDGE` bit set at one polarity but not the other. This leads to a difference in the reported number of false positives as we only consider those without the `EDGE` flag set.

The suspicion is that the differences in the precision of the centroids is linked to the differences in flags. We have not had the time to chase this down completely, but plan on doing so in Summer 2013.

12 Future Work

We intend to research additional aspects of the problem going forward, including:

- How the false detection rate varies with more realistic SEDs;

- How the false detection rate varies with airmass and passband;

- How the system behaves with a realistic mix of stars and galaxies;

- How the false detection rate looks in crowded fields;

- Investigate the propagation of covariances as a function of convolution and requirements on the template signal-to-noise.

- Adoption of Gaussian processes for spatial modeling of the kernel;

- Investigating the differences in measurement on positive/negative sources;

- Developing a measurement suite (shape, moment, goodness of fit) for sources in pre–filtered data and evaluating the photometric performance of the pre–filtered measurements.

- Determining how much "on the fly" configuration we may do to optimize PSF–matching.

Per Sensor MSE Comparison					
Visit	Pre–filter	N_{Same}	$N_{<3}$	ΔFP	$\max(\Delta FP)$
v6866601	True	0	2	22	4
	False	4	6	24	10
v6866602	True	3	5	7	2
	False	6	7	8	6
v6866603	True	6	7	3	1
	False	6	8	3	1

Table 7: We examine the configs chosen based on the minimum $\langle$MSE$\rangle$ and compare them to the config of the minimum number of false positives. The sample here consists of 9 sensors per visit. N_{Same} is the number of times the minimum $\langle$MSE$\rangle$ coincides with the minimum in false positives. $N_{<3}$ is the number of times the minimum false positive config is one of the top three configs when rank ordered by increasing $\langle MSE \rangle$ value. ΔFP is the total change in false positive count using the minimum $\langle$MSE$\rangle$ config as compared to the sum of the minimum false positives per sensor. Column $\max(\Delta FP)$ is the maximum change in detected false positives on any sensor in that visit.

WCS RMS in mas at various stages				
Sensor	Deep	Science	Deep to Science	Coadd to Science
R22:S00	4.2	5.8	3.8	382
R22:S10	2.6	4.4	4.4	389
R22:S20	1.4	3.6	4.2	375
R22:S01	2.6	5.0	4.2	86
R22:S11	1.2	3.4	4.0	81
R22:S21	1.6	3.8	4.0	382
R22:S02	1.6	3.4	4.0	178
R22:S12	1.8	4.0	4.0	78
R22:S22	1.4	4.2	4.2	389

Table 8: We looked at the resulting RMS from WCS fitting on the source lists of the deep exposure to the reference catalog (column 1), the science exposure to the reference catalog (col 2), fitting the deep exposure directly to the science exposure (col 3), and fitting a warped coadd to the science exposure (col 4). Fitting the coadd to the science exposure produces 1–2 orders of magnitude more scatter in the fit than at any other point.

References

Alard, C. 2000, *A&AS*, 144, 363

Alard, C., & Lupton, R. H. 1998, *ApJ*, 503, 325

Israel, H., Hessman, F. V., & Schuh, S. 2007, Astronomische Nachrichten, 328, 16

Kaiser, N. 2004, The Likelihood of Point Sources in Pixellated Images (Unpublished)

Price, P., & Magnier, E. 2010, Psf-Matching for Subtraction and Stacking (Unpublished)

Ulmer, W. 2010, Inverse Problems, 26, 085002

Appendices

A   Phosim Configuration for Production Data

```
zenith_v 23.0
qevariation 0.0
backbeta 4
raydensity 0.0
airglowvariation 0
cleartracking
clearperturbations
clearclouds
saturation 0
blooming 0
diffractionmode 0
centroidfile 1
```

B   Phosim Data for Extended Analysis

Images for the extended analysis contain stars and galaxies, at densities appropriate for high Galactic latitudes, with a range of spectral energy distributions, but with no asteroids, no variability, no cosmic rays, and no proper motions or parallax. Images simulated from these catalogs were generated using the input parameters described above, with the following exceptions.

Visit 11111111 was generated for the i-band:

- Visit 1111111100: A pair of 300s exposure template images with an atmospheric seeing value of 0.88 arcsec.

- Visit 1111111101: A pair of 15s exposure exposures with 0.6 arcsec atmospheric seeing at zenith.

- Visit 1111111102: A pair of 15s exposure exposures with 0.88 arcsec atmospheric seeing at zenith.

- Visit 1111111103: A pair of 15s exposure exposures with 1.2 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 1111111104: A pair of 15s exposure exposures with 0.6 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 1111111105: A pair of 15s exposure exposures with 0.88 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 1111111106: A pair of 15s exposure exposures with 1.2 arcsec atmospheric seeing at an altitude of 45 degrees.

Visit 22222222 was generated for the g-band:

- Visit 2222222200: A pair of 300s exposure template images with an atmospheric seeing value is 0.88 arcsec.

- Visit 2222222201: A pair of 15s exposure exposures with 0.6 arcsec atmospheric seeing at zenith.

- Visit 2222222202: A pair of 15s exposure exposures with 0.88 arcsec atmospheric seeing at zenith.

- Visit 2222222203: A pair of 15s exposure exposures with 1.2 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 2222222204: A pair of 15s exposure exposures with 0.6 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 2222222205: A pair of 15s exposure exposures with 0.88 arcsec atmospheric seeing at an altitude of 45 degrees.

- Visit 2222222206: A pair of 15s exposure exposures with 1.2 arcsec atmospheric seeing at an altitude of 45 degrees.