

**No jsMath TeX fonts found -- using image fonts instead.**  
 These may be slow and might not print well.  
 Use the jsMath control panel to get additional information.

[jsMath Control Panel](#)

[Hide this Message](#)

# ANDREW BECKER

RESEARCH ASSISTANT PROFESSOR  
 DEPARTMENT OF ASTRONOMY, UNIVERSITY OF WASHINGTON

## SECTIONS

[Home](#)  
[Curriculum Vitae](#)  
[Research](#)  
[Publications](#)  
[Teaching](#)  
[Diversity & Outreach](#)  
[Data Releases](#)

## HIGH ORDER TRANSFORM OF PSF AND TEMPLATE SUBTRACTION

This package is designed to photometrically align one input image with another, after they have been astrometrically aligned. This is an implementation of the Alard 1999 algorithm for image subtraction. This is very similar to the ISIS package, but with a variety of changes and (dare I say it again) improvements. The software has been applied in a variety of real-time data pipelines with great success, and much work has gone into making this a robust package.

## DESCRIPTION

The crux of the difference imaging problem is to find a convolution kernel  $K$  that matches the PSFs of two astronomical images,  $I$  (referred to as the image) and  $T$  (referred to as the template). These images in general are taken under different conditions, including atmospheric transparency, atmospheric seeing, or exposure times. One may even difference data taken from different sites and equipment entirely. However, this technique does **not** work well for differencing data taken in different filters - to the extent that instrumental response as a function of wavelength is different on different equipment, differencing data taken from different sites does require some care.

Mathematically, the equation you want to solve is the minimization of the function

$$\sum ([T \otimes K](x,y) - I(x,y))^2$$

by solving for the kernel  $K$ . If  $K$  can be decomposed onto basis

## SOFTWARE

[C/C++ code](#)  
[Python Code](#)

## COLLABORATION LINKS

[SuperMACHO](#)  
[SuperMACHO Alerts](#)  
[SuperMACHO Plone](#)

## ESSENCE

[ESSENCE Alerts](#)

ESSENCE Plone  
 Deep Lens Survey  
 LSST  
 UW's LSST Page  
 SDSS Supernova Survey  
 SDSS Local

functions, then this is a linear least-squares problem, and can be solved uniquely by matrix inversion. In this method, the kernel is decomposed into basis functions

$$K(u, v) = \sum_n A_n K_n(u, v)$$

$$K_n(u, v) = e^{-(u^2+v^2)/2\sigma_k^2} u^i v^j.$$

By default in **HOTPANTS**,  $n = 3$  and

$$\sigma_{k1} = 0.7 \text{ pixels}; i + j \leq 6$$

$$\sigma_{k2} = 1.5 \text{ pixels}; i + j \leq 4$$

$$\sigma_{k3} = 3.0 \text{ pixels}; i + j \leq 2$$

MISC  
 Pictures  
 Taikonaut Falls  
 Old Pages  
 IMA Championships

so you have a narrow gaussian that is allowed to vary to high spatial order, a broader middle gaussian varying a lower spatial order, and a wide gaussian that varies at low spatial order. I will emphasize that this spatial order is confined to the size of the kernel, which is by default 21 x 21 pixels.

These defaults are not optimal in the global sense that you really want to feed in the relative sizes of your PSFs so that you can estimate the scales at which they are expected to be different, and thus the size of the gaussians that make up your convolution kernel. **HOTPANTS** allows you to enter on the command line the sizes of the kernel gaussians and their spatial orders, as well as the size of the kernel. In practice, **HOTPANTS** is called by Perl or Python scripts that scale these as functions of the relative PSF sizes. However, **the way that HOTPANTS is implemented, these cannot be fit for.**

#### METHOD

**HOTPANTS** divides the image up into multiple **regions**. You fit for one convolution kernel within each region. An image might have multiple regions if e.g. the aspect ratio is large (4 x 1) and you can't expect a single function to model its spatial variation. Each region is divided up into **stamps**, generally 1 per every 100 pixels (a 2k x 4k image has 20 x 40 stamps). Within each stamp, you choose multiple **sub-stamps** that are centered on individual astronomical objects. The idea is that you want constraints on the kernel in each stamp, and you have multiple options within each stamp in case particular objects are sigma-clipped from the process. Since you are fitting for a function that is spatially varying, you want to ensure constraints across the entire image (especially in the corners).

Input masks may be fed to the code, which will allow it to avoid bad pixels while looking for sub-stamps. This allows it to distinguish between completely bad pixels which should be ignored (e.g. saturated pixels), pixels which you might trust but don't want to use while fitting for the kernel (e.g. interpolated pixels), and pristine untouched pixels. These masks are convolved and may be output from the software to be used in subsequent stages of the processing (e.g. diffim photometry) that are designed to take advantage of this information. In addition, noise maps may be fed to, and delivered from, the code. Because of the convolution, some of the noise will get lost in off-diagonal covariance terms. This is unavoidable, the best you can do is track the image noise so that you know the significance of detections in your difference image.

In practice, the program determines (or is fed a list of) objects to fill the sub-stamps. These objects are extracted from the template and image, and fed to the the matrix equations

$$M\mathbf{a} = \mathbf{B}$$

$$M_{ij} = \int [T \otimes K_i](x,y) \frac{[T \otimes K_j](x,y)}{\sigma(x,y)^2} dx dy.$$

$$B_i = \int I(x,y) \frac{[T \otimes K_i](x,y)}{\sigma(x,y)^2} dx dy.$$

Note that because of a constraint of constant flux scaling in the kernel across the image, the denominators  $\sigma(x,y)^2$  are not used in the matrix calculations, but are used to quantify the residuals in each sub-stamp.

For each object, this kernel is determined by solving for  $a$  above. Because some objects may be varying in brightness or position, the **kernel sum** is used as an initial way to sigma clip outliers from this distribution. Another figure of merit used to reject bad stamps is the average pixel residual (after convolution and template subtraction) divided by the estimated variance in that pixel (the true noise is not known until the final convolution kernel is determined). Another figure of merit that may be used is the width of the histogram of residuals in this difference stamp. **HOTPANTS** will choose which image to convolve based upon these figures of merit (deconvolution tends to leave larger residuals), unless the user demands otherwise.

Since the PSF varies spatially in all astronomical images, we must model  $K$  as a spatially varying function also. In practice, we use

$$K(u, v) = \sum_n a_n(x, y) K_n(u, v).$$

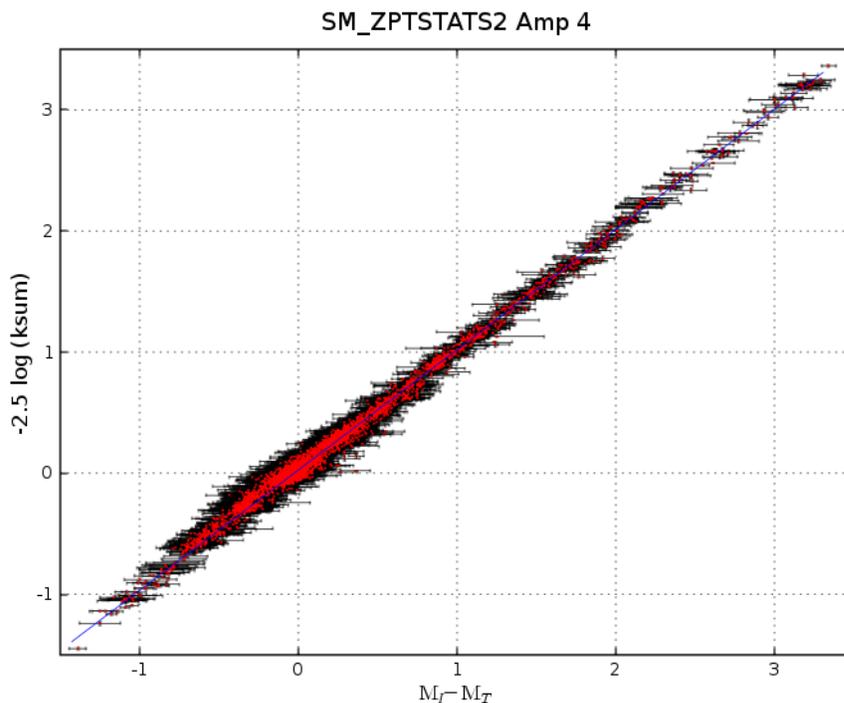
$$a_n(x, y) = \sum_{i,j} b_{ij} x^i y^j$$

but you could also use other sets of orthogonal polynomials to fit the spatial variation. Ideally, this should be informed by the known degree and shape of PSF variation in your focal plane.

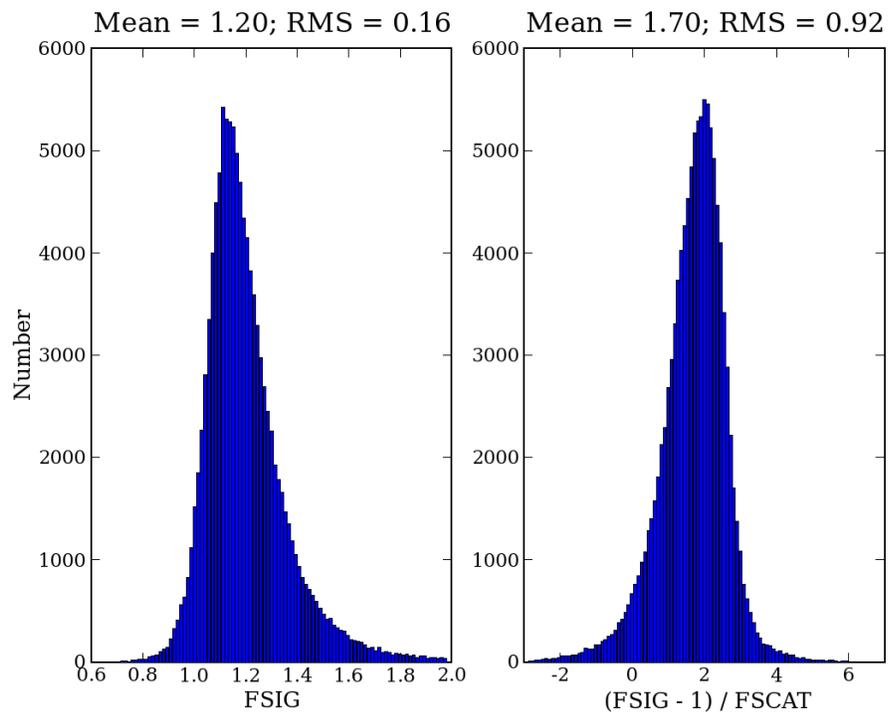
Since we have decomposed the spatial variation of the kernel using basis functions, we are able to solve for the spatial coefficients of the kernel with another set of linear equations. We have constraints on the kernel at the location  $x, y$  of each sub-stamp. We solve for the spatially varying coefficients using the set of all sub-stamp solutions. Outliers from the fit are sigma-clipped out, and neighboring sub-stamps from the stamp are brought into the fit in their place. This process repeats until no sub-stamps are clipped out.

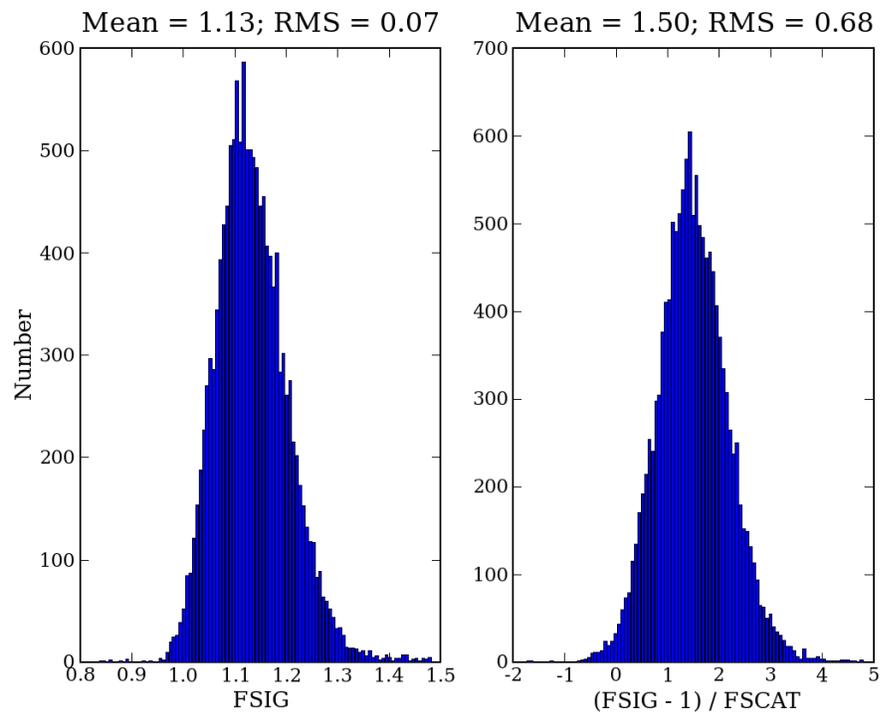
#### METRICS FOR SUCCESS

The photometric scaling of the two images comes out in the convolution **kernel sum**. Ideally, this should relate to the relative zero-points in the image through  $-2.5 * \log(\text{ksum}) = Z_{\text{pt}_I} - Z_{\text{pt}_T}$ . This plot is shown here for an ensemble of SuperMACHO data. The  $\chi^2/\text{dof}$  for this fit is very nearly 1.



The distribution of residuals  $(\text{diff}/\sigma)^2$  in each sub-stamp also provides an estimate of how good the overall fit is. A complication is that, because of convolution, some of the noise is lost in covariance, and thus the noise per pixel in the final image is an underestimate of the true noise. This biases this metric to values larger than 1.0. To demonstrate this, two sets of figures are shown below. For each, the plot on the left shows a histogram of FSIG, the average value across sub-stamps of the average sigma per pixel; on the right is how many sigma this value is from the expectation value (1.0) using the RMS of the distribution across sub-stamps (FSCAT). 5-sigma outliers are clipped. At the top is the plot for 100k SuperMACHO difference images; at the bottom is the same for 14k ESSENCE difference images.





#### COLLABORATIONS THAT USE HOTPANTS

- Dark Energy Survey (?)
- Palomar Transient Factory
- SuperMACHO
- ESSENCE
- DLS
- SDSS-II Supernova Search
- MCELS
- CHASE

#### IMPROVEMENTS OVER ISIS

- Use of pixel masking to avoid bad stamps
- Propagation of pixel masks for photometry packages
- Propagation of noise maps for photometry packages
- Freedom to choose number, shape, and degree of Gaussian kernel components
- Sigma clipping of bad sub-stamps; replacement with neighboring sub-stamps
- Input lists of objects to use to constrain the kernel
- Software may choose which image to convolve

#### FUTURE IMPROVEMENTS

- How to deal with differential atmospheric refraction (both in the remapping and in the diffim kernel)

- Make a more generalized kernel (delta functions instead of Gaussians)
- Allow user to send / choose basis functions
- Integrate more closely with PSF modeling routines
- Make multi-threaded

LAST MODIFIED 2009-07-01T10:05:08BY BECKER@ASTRO.WASHINGTON.EDU