

Wide-field Infrared Survey Explorer



Subsystem Design Specification

A WISE Astronomical Image Co-adder (AWAIC)

Version 3.1, 15-July-2008

Prepared by: Frank Masci



**Infrared Processing and Analysis Center
California Institute of Technology**

WSDC D-D005

Concurred By:

Roc Cutri, WISE Science Data Center Manager

Tim Conrow, WISE Science Data Center Lead Architect

Frank Masci, WISE Science Data Center Cognizant Engineer/Scientist

Revision History

Date	Version	Author	Description
August 20, 2007	1.0	Frank Masci	Initial Draft
November 1, 2007	2.6	Frank Masci	Optimized when creating a simple (unHIRES'd) co-add.
November 2, 2007	2.7	Frank Masci	New command-line inputs: -rf, -ct, -if.
November 29, 2007	2.8	Frank Masci	Made specification of output co-add pixel scale flexible.
February 12, 2008	2.9	Frank Masci	<ul style="list-style-type: none"> * Allow specification of output <i>square</i> co-add pixel scales in absolute arcsec units (-pa); * Read in bad-pixel FITS image masks with up to 32-bit integers per pixel; * Allow specification of a bit-string template for conditional flagging of pixels (-m); * Replaced bilinear interpolation option with a more robust area-overlap weighting method (-if).
April 16, 2008	3.0	Frank Masci	<ul style="list-style-type: none"> * Added option (-sc) to perform "overlap-area" weighted co-add; * Option to generate standard deviation co-add of input pixel stacks (-o4) for -sc case.
July 10, 2008	3.1	Frank Masci	* Replaced cfitsio header reading function with ffhdr2str function (this properly stores all WCS and distortion keys).

Table of Contents

1. INTRODUCTION.....	6
1.1 Purpose and Scope	6
1.2 Document Organization.....	6
1.3 Applicable Documents	6
1.4 Requirements	7
1.5 Acronyms.....	8
2 OVERVIEW	9
2.1 HIRES Functionality.....	10
3 INPUT/OUTPUT SPECIFICATION.....	10
3.1 Inputs.....	10
3.2 Outputs.....	13
4 CO-ADD SUBSYSTEM AND PRE-PROCESSING	14
4.1 CO-ADD Subsystem Overview.....	14
4.2 Instrumental Calibration and Bad Pixel Masks.....	14
4.3 Outlier Detection.....	15
4.4 Background Matching	15
5 AWAIC PHILOSOPHY.....	15
6 AWAIC PROCESSING	16
6.1 Assumptions and Advisories.....	16
6.2 Processing Phases.....	20
6.3 Algorithm (the AWAIC brain)	21
6.4 Interpolation onto the “cell-pixel” grid	25

6.5	Pixel Overlap-Area Weighted Interpolation	29
6.6	Command-line Usage Examples	31
7	EXAMPLES AND TESTING	32
8	QUANTITATIVE ANALYSIS	32
8.1	Co-add Pixel Noise	34
8.2	Aperture Photometry and Correlated Noise.....	35
9	WORK IN PROGRESS AND PLANS	40
10	REFERENCES	41

1. INTRODUCTION

1.1 Purpose and Scope

This Subsystem Design Specification (SDS) document describes the basic requirements, assumptions, definitions, software-design details, algorithm and necessary interfaces for the CO-ADD subsystem of the WISE Science Data System (WSDS). It will be used to trace the incremental development of this subsystem, and contains sufficient detail to allow future modification or maintenance of the software by developers other than the original developer. This document is an evolving document as changes may occur in the course of science instrument hardware design and maturity of operational procedures.

This document focuses on one component (module) of the CO-ADD subsystem - AWAIC. It represents the primary module of this subsystem. Its purpose is to read in multiple image-frame exposures within a pre-defined region on the sky, and combine them in an optimal manner. It will be used to generate the WISE Image Atlas. Two other modules of the CO-ADD subsystem which mesh closely with (and executed prior to) AWAIC are: *outlier detection* and *background matching*. Specific design details for these modules are described in separate SDS documents (see for §1.3 for details).

1.2 Document Organization

This document is organized along the major themes of Requirements; Other Software Interfaces; Assumptions; Functional Descriptions and Dependencies; Input/Output; Algorithm Descriptions; Testing; and Major Liens.

The material contained in this document represents the current understanding of the capabilities of the major WISE systems and sub-systems. Areas that require further analysis are noted by TBD (To Be Determined) or TBR (To Be Resolved). TBD indicates missing data that are not yet available. TBR indicates preliminary data that are not firmly established and subject to change.

1.3 Applicable Documents

- WISE Project Plan (Level 1 Requirements)
- WISE Science Requirements Document (Level 1.5 Requirements)
- WSDC Functional Requirements Document *WSDC D-R001* (FRD – Level 4 Requirements):
http://web.ipac.caltech.edu/staff/roc/wise/docs/WSDC_Functional_Requirements_all.pdf
- WSDS Functional Design Document *WSDC D-D001* (FDD)

- WSDC Software Management Plan *WSDC D-M002* (SMP):
<http://web.ipac.caltech.edu/staff/roc/wise/docs/wsdc-smp-draft.pdf>
- WSDC Science Data Quality Assurance Plan *WSDC D-M004* (QAP):
http://web.ipac.caltech.edu/staff/roc/wise/docs/QA_Plan_WSDC_2007-03-01.pdf
- Software Interface Specification (SIS) *WSDC D-I101* – Frame Processing Mask:
<http://web.ipac.caltech.edu/staff/fmasci/home/wise/InstruCal01.txt>
- Software Interface Specification (SIS) *WSDC D-I102* – Frame WCS FITS Header
Keywords: <http://web.ipac.caltech.edu/staff/fmasci/home/wise/SFPWrap01.txt>
- Proposed WISE Image Atlas Specifications (reviewed at Oct '07 Science Team meeting):
http://web.ipac.caltech.edu/staff/fmasci/home/wise/Atlas_image_spec_v1.2.pdf
- Frame Co-addition Peer Review presentation (11/15/2007):
http://web.ipac.caltech.edu/staff/fmasci/home/wise/Co-addition_PeerReview.pdf
- Frame Co-addition Peer Review Report *WSDC D-A001* (summary of 11/15/2007 Peer Review):
http://spider.ipac.caltech.edu/staff/fmasci/home/wise/awaic_peerreview_report.pdf
- Frame Pipeline Processing: Instrumental Calibration (internal working document):
http://web.ipac.caltech.edu/staff/fmasci/home/wise/ScanPL_instrumental_cal.pdf
- Frame Co-addition Critical Design Review (01/30/2008):
http://web.ipac.caltech.edu/staff/fmasci/home/wise/Co-addition_CDRJan08.pdf
- Subsystem Design Specification (SDS) *WSDC D-D014* – AWOD: A WISE Outlier Detector: <http://web.ipac.caltech.edu/staff/fmasci/home/wise/sds-wsdc-D014-awod.pdf>
- Subsystem Design Specification (SDS) *WSDC D-D???* – *Bmatch: Background Matcher* (in preparation).
- Subsystem Design Specification (SDS) *WSDC D-D???* – *Frame Co-adder Wrapper Script* (in preparation).

1.4 Requirements

Below we summarize the requirements pertaining to the format and properties of the final release WISE Atlas Image products. These are from the WSDC Functional Requirements Document (§1.3).

- *L4WSDC-001*: The WSDC shall produce a digital Image Atlas that combines multiple survey exposures at each position on the sky.
- *L4WSDC-021*: The images in the final WISE Image Atlas shall be re-sampled to a common pixel grid at all wavelengths.
- *L4WSDC-022*: The photometric calibration of the final WISE Image Atlas shall be tied to the photometric calibration of the final WISE Source Catalog.
- *L4WSDC-023*: The WSDC shall make all WISE image data available in accordance to the Flexible Image Transport (FITS) astronomical data standard.
- *L4WSDC-026*: The WSDC shall generate and archive coverage maps that show the number of independent observations that go into each pixel of the Image Atlas images in each band. The coverage numbers shall take into account focal plane coverage and losses due to poor data quality, low responsivity and/or high noise masked pixels, and pixels lost because of cosmic rays and other transient events.
- *L4WSDC-051*: The WSDC shall make the WISE catalog and image products available to the community via the internet through appropriate web-based tools.
- *L4WSDC-053*: The WSDC shall make the Image Atlas and Catalog products accessible to the astronomical community in collaboration with the NASA/IPAC Infrared Science Archive (IRSA) to ensure long-term availability beyond the end WISE missions operations and data processing phase, and to insure interoperability with other NASA mission archives.
- *L4WSDC-060*: The WSDC archive shall provide a web-based interface to enable selection, display and retrieval of any or all single-epoch images and combined Atlas Images based on position or time of observation for the purpose of quality assurance, validation and analysis. The goal shall be to select on any image metadata parameter.
- *L4WSDC-078*: The WISE science data products shall use the International Celestial Reference System (ICRS) to describe the positions and motions of celestial bodies. WISE astrometry shall be mapped into the ICRS using the 2MASS All-Sky Point Source Catalog as the primary astrometric reference.

1.5 Acronyms

AWAIC	A WISE Astronomical Image Co-adder
AWOD	A WISE Outlier Detector
CPU	Central Processing Unit
CR	Cosmic Ray
CVZ	Continuous Viewing Zone
DN	Data Number
FDD	Functional Design Document
FRD	Functional Requirements Document
FITS	Flexible Image Transport System
FOV	Field of View
FPA	Focal Plane Array
FRESCO	Full RESolution CO-add
FWHM	Full Width at Half Maximum

HIRES	High Resolution
HST	Hubble Space Telescope
I/O	Input / Output
IPAC	Infrared Processing and Analysis Center
IRAC	Infra-Red Array Camera
IRAS	Infra-Red Astronomical Satellite
IRSA	NASA/IPAC Infra-Red Science Archive
MAD	Median Absolute Deviation
MCM	Maximum Correlation Method
MFPREX	Multi Frame Pointing REconstruction
MOPEX	[SSC's] MOsaicker and Point source Extractor
NEP	North Ecliptic Pole
PA	Position Angle
PRF	Point Response Function
PSF	Point Spread Function
SDS	Subsystem Design Specification
SEP	South Ecliptic Pole
SFPREX	Single Frame Pointing REconstruction
SIP	Simple Imaging Polynomial
SIS	Subsystem Interface Specification
SMP	Software Management Plan
SNR	Signal to Noise Ratio
SSC	<i>Spitzer</i> Science Center
TBD	To Be Determined
TBR	To Be Resolved
2MASS	Two Micron All Sky Survey
QA	Quality Assurance
QAP	Quality Assurance Plan
WCS	World Coordinate System
WISE	Wide-field Infrared Survey Explorer
WSDC	WISE Science Data Center
WSDS	WISE Science Data System
WST	WISE Science Team

2 OVERVIEW

WISE shall produce image data frames consisting of 1024×1024 pixels for bands 1, 2 and 3 with a projected size of 2.75 arcsec/pixel, and 512×512 pixels for band 4 with a size of 5.5 arcsec/pixel. This corresponds to image dimensions of $\approx 47 \times 47$ arcmin on the sky for all bands. Prior to co-adding and mosaicing, the image frames are first processed to remove instrumental signatures, their WCS refined using an astrometric catalog, and then a photometric zero-point is inserted into their headers to represent the photometric calibration. They are then ready for the multi-frame pipeline (described in the FDD). This consists of several modules, one of which is

the image co-adder/mosaicker called AWAIC. This document describes the algorithm, philosophy and usage behind AWAIC.

Inputs to AWAIC are a list of FITS image file names, a list of FPA position dependent PRFs, optional uncertainty and bad-pixel mask images, and processing parameters. Image construction in AWAIC is based on using the PRF as an interpolation kernel, although it is generic enough to take any kernel as input. A straight overlap-area weighted interpolation option is also available. The primary outputs from AWAIC are an intensity co-add image, an associated depth-of-coverage map denoting effectively the number of “unmasked” PRF contributions at points on the sky, and optionally, an uncertainty (sigma) image. AWAIC is written in ANSI/ISO C.

2.1 HIRES Functionality

We mention that AWAIC is also capable of resolution enhancement using an iterative (Richardson-Lucy like) procedure. However, this is *not in the WSDS baseline design*, and will not be performed in automated operations. The WISE Atlas image products (deliverables) will be simple co-adds based on a PRF-interpolation scheme. In fact, the very first iteration of the AWAIC algorithm results in a PRF-interpolated co-add. Further iterations imply resolution enhancement. If desired, resolution-enhanced products will only be made offline. This is because much hands-on analysis and tuning is needed for optimal enhancement, e.g., in the construction of PRFs at different FPA locations and the removal of artifacts. Also, the iterative procedure itself is CPU intensive.

3 INPUT/OUTPUT SPECIFICATION

3.1 Inputs

AWAIC takes all of its input from the command-line, which is set up by a startup script and controlled by the WSDS pipeline executive, or, it can be set up manually and executed standalone. Prior to parsing the command-line inputs, default values for the optional input parameters are assigned. Table 1 summarizes all command-line inputs, their purpose and default assignments. Some example command-line usage cases are given in §6.5.

Variable name	option	Description	Data-type / Format	Units	Default
inp_image_list_fname	-f1	List of image frames in 32-bit floating point FITS format.	Char*256 (text file)	Null	Required input.
inp_mask_list_fname	-f2	List of bad-pixel masks in 32-bit integer FITS format. Only integer values $0 \rightarrow 2^{31}$ are stored.	Char*256 (text file)	Null	No masking - all frame pixels used.
inp_uncert_list_fname	-f3	List of uncertainty images in 32-bit floating point FITS format.	Char*256 (text file)	Null	No variance weighting done and no output sigmas

					computed.
inp_prf_list_fname	-f4	List of PRF images in 32-bit floating point FITS format labeled with FPA position.	Char*256 (text file)	Null	Required input.
inp_fatalmask_bits	-m	Bitstring template for conditional flagging of pixels defined in masks. Max. value allowed = $2^{31} = 2147483647$.	I*4 unsigned int	Null	0 \Rightarrow no masking.
mosaic_size_x	-X	E-W mosaic dimension for crota2=0.	R*4 float	degrees	Required input.
mosaic_size_y	-Y	N-S mosaic dimension for crota2=0.	R*4 float	degrees	Required input.
RA_center	-R	RA of mosaic center.	R*4 float	degrees 0...360	Required input.
Dec_center	-D	Dec. of mosaic center.	R*4 float	degrees -90...+90	Required input.
mosaic_rotation	-C	Desired mosaic rotation in terms of crota2: +Y axis W of N.	R*4 float	degrees 0...360	0.0
pixelscale_factor	-ps	Ratio of mosaic linear pixel size to input pixel size: same in X and Y.	R*4 float	Null	0.5 (min allowed = 0.1)
pixelscale_absolute	-pa	Pixel scale of output mosaic in absolute units: same in X and Y.	R*4 float	arcsec	If not specified, -ps <input> is used.
mos_cellsize_factor	-pc	Ratio of internal cell grid pixel size to mosaic pixel size. For PRF placement and interpolation. Cell pixel size must be same as input PRF pixel size.	R*4 float	Null	0.5 (min allowed = 0.2).
pixelflux_scale_flag	-sf	Scale output co-add pixel flux with pixel size? 0=no; 1=yes. E.g., if inputs are surface brightness, then want 0. If DN, then want 1.	I*2 int	Null	0
simple_coadd_flag	-sc	Create simple co-add/mosaic using exact overlap-area weighting: 1=yes, 0=no.	I*2 int	Null	0
num_mcm_iterations	-n	Number of MCM iterations for HIRES.	I*2 int	Null	1 (simple co-add, no HIRES).
rotate_prf_proj_flag	-rf	Rotate PRF when projecting input frame pixels? 1=yes, 0=no. Recommended for HIRES ($n > 1$).	I*2 int	Null	0 (not needed for simple co-add).
prf_cell_size_tol	-ct	Maximum tolerance for difference between cell-	R*4 float	arcsec	0.0001

		grid pixel size [-pc] and input PRF pixel size.			
interpolation_option	-if	Interpolation method for PRF onto cell-pixel grid: 0=nearest neighbor, 1=area-overlap weighting.	I*2 int	Null	0
out_mosaic_image	-o1	output mosaic image FITS filename.	Char*256 (text file)	Null	Required input.
out_mosaic_cov_map	-o2	output mosaic coverage map FITS filename.	Char*256 (text file)	Null	Required input.
out_uncert_mosaic	-o3	output uncertainty mosaic FITS filename.	Char*256 (text file)	Null	No uncertainty (sigma) mosaic generated.
out_stddev_mosaic	-o4	output standard deviation mosaic FITS filename. Only possible if “-sc 1” specified.	Char*256 (text file)	Null	No standard deviation mosaic generated.
Debug Flag	-g	switch to print debug statements to stdout.	Null	Null	0
Verbose Flag	-v	switch to print details of processing to stdout.	Null	Null	0

Table 1: Command-line inputs and options

When AWAIC is executed with no command-line inputs, or, with a single “-help” (e.g., as “awaic -help”), a command-line synopsis and tutorial is printed on the screen. This is reproduced below:

Program aWaic: A WISE Astronomical Image Coadder, Version 3.1

Usage: awaic

```

-f1 <inp_image_list_fname> (Required; list of images in FITS format)
-f2 <inp_mask_list_fname>  (Optional; list of bad-pixel masks in 32-bit INT
                           FITS format; only values 0 -> 2^31 are used)
-f3 <inp_uncert_list_fname> (Optional; list of uncertainty images in
                           FITS format)
-f4 <inp_prf_list_fname>   (Required; list of PRF FITS images each
                           labeled with location on array)
-m <inp_fatalmask_bits>   (Optional; bitstring template specifying
                           pixels to flag as set in input masks; Default=0)
-X <mosaic_size_x>        (Required [deg]; E-W mosaic dimension
                           for crota2=0)
-Y <mosaic_size_y>        (Required [deg]; N-S mosaic dimension
                           for crota2=0)
-R <RA_center>            (Required [deg]; RA of mosaic center)
-D <Dec_center>           (Required [deg]; Dec. of mosaic center)
-C <mosaic_rotation>      (Optional [deg]; in terms of crota2:
                           +Y axis W of N; Default=0)
-ps <pixelscale_factor>   (Optional; output mosaic linear pixel scale
                           relative to input pixel scale; Default=0.5)
-pa <pixelscale_absolute> (Optional [asec]; output mosaic pixel scale
                           in absolute units; if specified, over-rides -ps)
-pc <mos_cellsize_factor> (Optional; for PRF placement: internal linear
                           cell pixel size relative to mosaic pixel size;

```

```

      =input PRF pixel sizes (-f4); Default=0.5)
-sf <pixelflux_scale_flag> (Optional; scale output pixel flux with pixel
      size: 1=yes, 0=no; Default=0)
-sc <simple_coadd_flag> (Optional; create simple co-add/mosaic using exact
      overlap-area weighting: 1=yes, 0=no; Default=0)
-n <num_mcm_iterations> (Optional; number of MCM iterations;
      Default=1 => coadd, no resolution enhancement)
-rf <rotate_prf_proj_flag> (Optional; rotate PRF when projecting input
      frame pixels: 1=yes, 0=no; Default=0;
      recommended for >1 MCM iterations)
-ct <prf_cell_size_tol> (Optional [asec]; maximum tolerance for difference
      between cell-grid pixel size [-pc] and input
      PRF pixel size; Default=0.0001 arcsec)
-if <interpolation_option> (Optional; method for interpolating PRF onto
      co-add cell-grid: 0=nearest neighbor,
      1=area-overlap weighting; Default=0)
-o1 <out_mosaic_image> (Required; output mosaic image FITS filename)
-o2 <out_mosaic_cov_map> (Required; output mosaic coverage map
      FITS filename)
-o3 <out_uncert_mosaic> (Optional; output uncertainty mosaic
      FITS filename)
-o4 <out_stddev_mosaic> (Optional; output standard deviation mosaic
      FITS filename; only possible under -sc 1)
-g (Optional; switch to print debug statements
      to stdout and files)
-v (Optional; switch to print more verbose output)

```

3.2 Outputs

There are four possible co-add products from AWAIC, two of which are optional: The main intensity co-add, an associated coverage image map denoting effectively the number of “unmasked” PRF contributions at points on the sky represented on the co-add, and optionally, an uncertainty (sigma) co-add and stack standard-deviation image. All products are in FITS format with pixel values in IEEE single precision floating point (BITPIX = -32). All output images have the same dimensions and pixel size determined by the output/input pixel scale ratio (-ps <pixelscale_factor>) or absolute output pixel size (-pa <pixelscale_absolute>), and the desired mosaic linear size on the sky: -X <mosaic_size_x> and -Y <mosaic_size_y>.

The uncertainty image contains a 1-sigma error estimate in the co-added signal for every pixel. These uncertainties implicitly contain the result of the full error-propagation from the instrumental calibration pipeline, as initiated using an error-model. For more details, see the algorithm description in §6.3.

The units of the pixel values in the output intensity co-add and corresponding uncertainty image reflect the input image units and can be scaled via the -sf <pixelflux_scale_flag> parameter. If the input image units are in Data Number (DN or counts), one may want to re-scale these according to pixel size so that the total counts in an aperture are conserved. The baseline plan for WISE is to have all co-add pixel units in native DN. The reason is that this convention becomes independent of any assumed calibration conversion factor. It will be computationally expensive to update all Atlas Image co-add pixels (and source extraction records) for changes in conversion factors from the photometric calibration. Instead, we shall store any calibration

conversion factor or offset as a single header keyword, e.g., MAGZP. This will be applicable to all pixels of an Atlas Image. This follows the 2MASS convention. For more details, see the *WISE Image Atlas Specifications* document (§1.3).

Along with the output image products, more information on details of processing can be written to standard output by setting the `-v` and `-g` switches. A processing log with all I/O is written at the end. Quality assurance metrics are generated by a separate module in the CO-ADD subsystem wrapper and are described in the *Frame Co-adder Wrapper Script* SDS document (referenced in §1.3).

4 CO-ADD SUBSYSTEM AND PRE-PROCESSING

4.1 CO-ADD Subsystem Overview

Figure 1 shows the proposed major processing steps in the CO-ADD subsystem. Note that algorithms for the outlier detection, background matching, and QA steps are described in separate SDS documents (see references in §1.3).

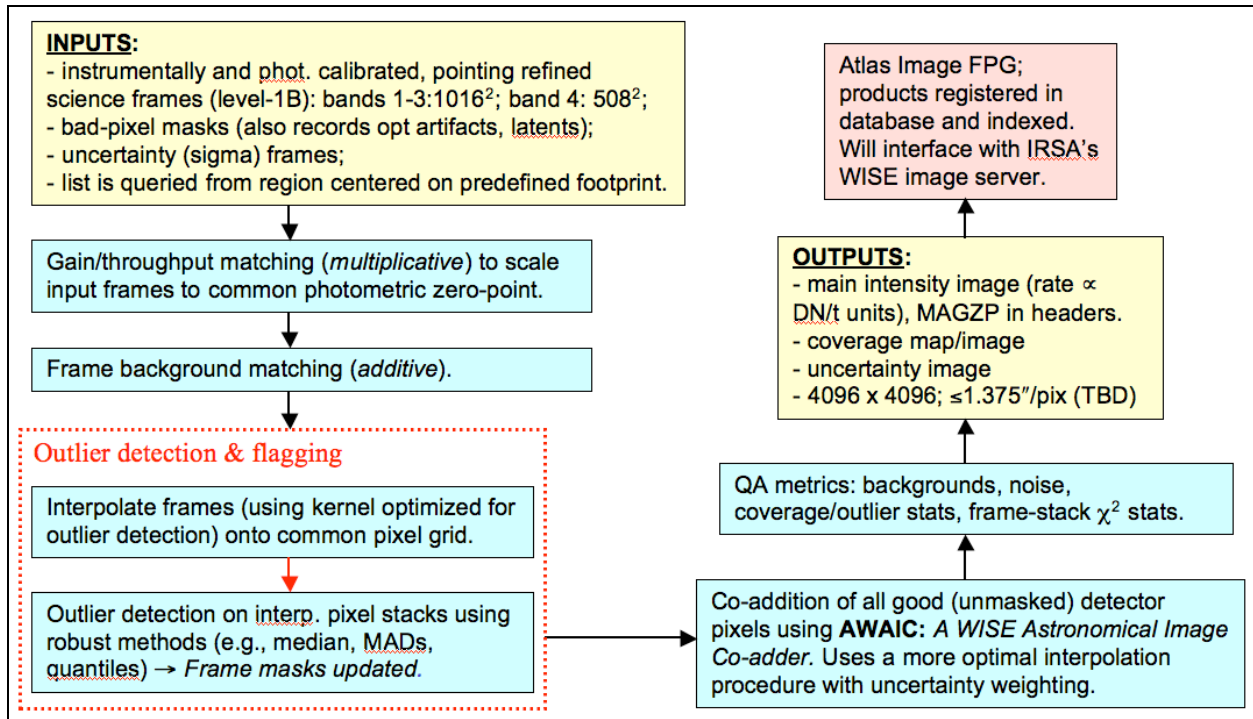


Figure 1: CO-ADD subsystem processing flow

4.2 Instrumental Calibration and Bad Pixel Masks

Prior to co-adding and mosaicing, image data frames are first processed to remove instrumental signatures. This is performed by the *instrumental calibration* subsystem (see *Frame Pipeline Processing* document in §1.3), and consists of the standard corrections: dark/bias subtraction, droop/electronic artifact removal (in specific bands), linearization, and flat-fielding. This pipeline also initializes pixel uncertainties using an error model for updating downstream. It also records and tracks “bad” pixels in a processing bit-mask to be updated later for optical artifacts (e.g., latents) and then propagated to the co-adder. AWAIC enables one to specify which “bad-pixel” bit-patterns to avoid and mask against prior to co-addition (input: `-m <inp_fatalmask_bits>`). Following instrumental calibration, the WCS of each frame is refined using an astrometric catalog (SFPReX and MFPreX subsystems), and then a photometric zero-point is inserted after calibration against a standard star network.

4.3 Outlier Detection

We take advantage of the redundancy from multiple frame exposures and attempt to detect and flag outliers (or inconsistencies) in the temporal domain. Details are described in the AWOD SDS document (see reference in §1.3).

4.4 Background Matching

The goal of background matching is to obtain seamless (and/or smooth) transitions between frames near their overlap regions in a co-add. We want to equalize background levels on frame-to-frame scales, but still preserve natural background variations and structures as much as possible. The bulk of variation in background levels will be due to instrumental transients and changing thermal environments. Details (including the prior gain/throughput matching step) are described in the *Bmatch* SDS document (see reference in §1.3).

5 AWAIC PHILOSOPHY

Very briefly, AWAIC is based on using the detector's Point Response Function (PRF) as an interpolation kernel when re-projecting and re-estimating pixels in an up-sampled co-add grid. Conventional image co-adders like IPAC's *Montage* and MOPEX tools are based on simple overlap-area-weighted averaging. In fact, PRF-interpolation or "PRF-weighted" averaging implicitly reduces to an area-weighted method when a detector's PRF is approximated by a top-hat (flat) function spanning ~ 1 detector pixel. This is also true if the telescope's optical Point Spread Function (PSF) is under-sampled by a detector. In this case, the PRF is close to top-hat anyway and area-weighted interpolation/averaging is the *only* way to proceed.

So, why use the PRF as an interpolation kernel? There are three reasons: The first is that the detector's PRF represents the real transfer function from which sky flux ("the truth") propagates through the measurement process to yield the observations. Other mock interpolation kernels (overlap-area weighting, sinc etc..) are mere approximations. These usually yield simple (and aesthetic) co-adds, but don't make use of the full spatial information collected by a detector's pixel (i.e., its spatial response) *if it happens to over-sample the optical PSF of a telescope*. The

second reason is that use of a PRF kernel directly allows for resolution enhancement if one decides to de-convolve its effects from the data (e.g., through a Richardson-Lucy process). Even if one has a detector which *under-samples* the PSF, one can still gain spatial resolution from iterative methods if multiple overlapping (randomly dithered) observations are available. The random dithering, in effect, reconstructs the PRF. The third reason is that the output intensity and uncertainty co-adds can be combined into a Signal-to-Noise Ratio (SNR) image to define the *most optimal* matched filter for *point* source detection. Use of the PRF as an interpolation kernel also leads to smoothing of an image (through a convolution). High frequency noise is smoothed out without affecting the point source signal sought for. Therefore the SNR of peaks is maximized. This will benefit processing at the WSDC since a source catalog is one of its release products.

The method used in AWAIC is very similar to that used by the 2MASS co-adder. The main difference is that the 2MASS co-adder assumes an analytic function for its interpolation kernel. This is based on a 2D Gaussian fit to the PSF (seeing profile) convolved with a square pixel response. This function was optimized such that "shape-bias" from the "square" detector pixels is minimized (smoothed out), but not so much that excessive variance is introduced from over-smoothing. Also, since the seeing PSF was under-sampled by the 2MASS detectors, the assumed interpolation kernel played more the role of a "smoothing kernel", e.g., for reducing pixel-shape bias. The direct use of a measured PRF as the interpolation/smoothing kernel in 2MASS co-addition was not possible due to the unstable and stochastic nature of the seeing. It's important to note that AWAIC can also mimic the 2MASS method in that it allows for any interpolation kernel to be specified, whether it be the actual PRF, or, an analytic function fit.

6 AWAIC PROCESSING

6.1 Assumptions and Advisories

Below we list the assumptions pertaining to the format, size and content of the image inputs. Many of these are checked internally by the program. If not satisfied, the program aborts with a message and a non-zero exit status written to *standard error*.

- The input lists of intensity images, masks, and uncertainty images must all have the same number of filenames listed in one-to-one correspondence.
- All image inputs are in FITS format.
- All input intensity, mask and uncertainty images are expected to have the same native pixel scale (but $\Delta X \neq \Delta Y$ is allowed); the same projection type (CTYPE header keywords); the same NAXIS1, NAXIS2 values (with NAXIS1 \neq NAXIS2 allowed); and the same EQUINOX.
- If optical-distortion information is available, this must be represented in the FITS headers of the intensity images using the Simple Imaging Polynomial (SIP) convention with the WCS keywords CDELT1, CDELT2, CROTA2 encoded in CD matrix format. For an example and more information, see SIS document *WSDC D-1102* referenced in §1.3.

- It is recommended that all image pixel scales (either from CDELTA or CD keywords) be represented in the FITS headers to at least 8 significant figures.
- The only projection types recognized by the software are: TAN, SIN, ZEA, STG and ARC. These are specific to the fast plane-to-plane projection algorithm used.
- The input list of PRF images must all be of the same size (i.e., their NAXIS1, NAXIS2 values the same) and have the same pixel scale (CDELTA1, CDELTA2 keywords). An example of a minimal FITS header for an input PRF is as follows (where the CDELTA* and FPALOC* values are not realistic).

```

SIMPLE      =                      T / file does conform to FITS standard
BITPIX      =                     -32 / number of bits per data pixel
NAXIS       =                      2 / number of data axes
NAXIS1      =                     <Num> / length of data axis 1
NAXIS2      =                     <Num> / length of data axis 2
CRPIX1      =    <0.5*(Num + 1)> / reference pixel for axis 1
CRPIX2      =    <0.5*(Num + 1)> / reference pixel for axis 2
CTYPE1      =      'RA--SIN' / projection type for axis 1
CTYPE2      =      'DEC--SIN' / projection type for axis 2
CDELTA1     =    -0.00019097222 / axis 1 scale [deg/pix]
CDELTA2     =     0.00019097222 / axis 2 scale [deg/pix]
FPALOCX     =         508.0 / center x coord of grid square
FPALOCY     =         711.2 / center y coord of grid square

```

- The values of the CTYPE1, CTYPE2 keywords in the PRF headers must be the same as those in the headers of the input intensity images.
- The PRF CDELTA1, CDELTA2 (pixel scale) values must be within some maximum tolerance of the internal co-add (sub-pixel) cell scales. I.e., for either the *X* or *Y* PRF pixel scale, they must satisfy:

$$|\text{pixelscale_factor} * \text{mos_cellsize_factor} * \text{input_image_CDELTA} - \text{input_prf_CDELTA}| \leq \text{prf_cell_size_tol}.$$

From left to right, the variables correspond to: the output mosaic-to-input image pixel size ratio (command-line parameter `-ps`), the mosaic *cell*-to-output mosaic pixel size ratio (command-line parameter `-pc`), the input image CDELTA1 or CDELTA2, and the maximum desired tolerance for this difference (command-line parameter `-ct <in arcsec>`). An easier way to get it right is to use the absolute output mosaic pixel scale parameter: `-pa <pixelscale_absolute in arcsec>`. If specified, this parameter will over-ride the `-ps` (mosaic-to-input pixel ratio) parameter. The PRF CDELTA1 and CDELTA2 must then satisfy:

$$|\text{pixelscale_absolute} * \text{mos_cellsize_factor} - \text{input_prf_CDELTA}| \leq \text{prf_cell_size_tol}.$$

Therefore, it is wise to first pick an *absolute* output mosaic pixel scale (`-pa`) and a cell size factor (`-pc`) before computing the necessary pixel scale of the input PRF(s).

- The FPALOCX and FPALOCY keywords in the PRF headers (see example header above) specify the location, in the native *X*, *Y* coordinate system of an input frame at which the PRF applies. This is for cases where the PRF is non-isoplanatic (i.e., varies over the FPA). The values of these keywords refer to the center coordinates of a square over which the PRF applies. The specific square regions are defined beforehand by partitioning the FPA into an $n \times n$ grid, with a PRF derived in each. The total number of

input PRFs is therefore n^2 . For a grid square labeled by integer coordinates (i, j) where $1 \leq i \leq n$ and $1 \leq j \leq n$ (see Figure 2), it will have physical center coordinates of:

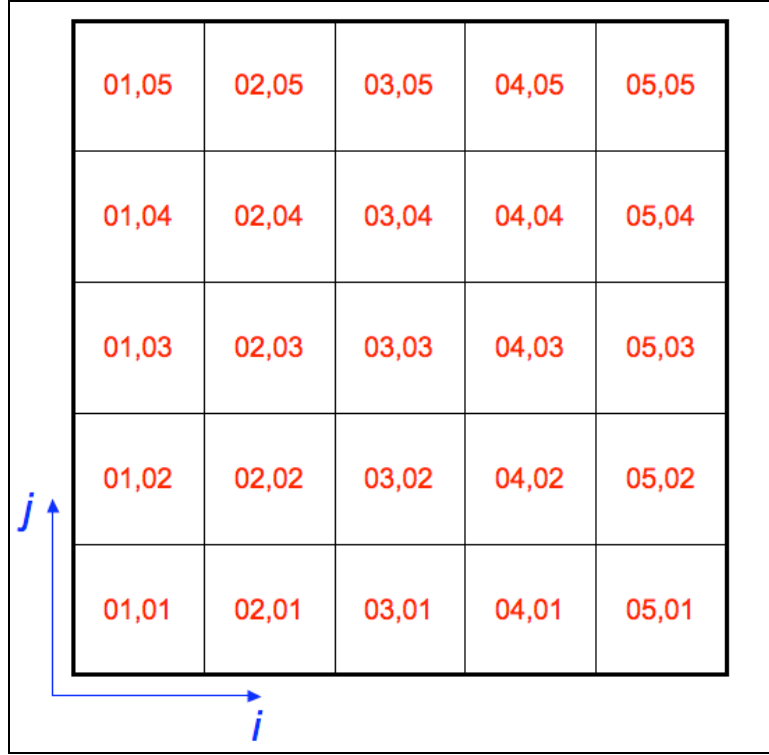
$$\text{FPALOCX} = \left(\frac{\text{NAXIS1}_f}{n} \right) \left(i - \frac{1}{2} \right);$$

$$\text{FPALOCY} = \left(\frac{\text{NAXIS2}_f}{n} \right) \left(j - \frac{1}{2} \right),$$

where $(\text{NAXIS1}_f, \text{NAXIS2}_f)$ refer to the dimensions of an input image frame. So for example, a grid square located at partition $(i, j) = (03, 04)$ in a 5×5 grid (see Figure 2) will have in the PRF header: $\text{FPALOCX} = (1016/5) * (3 - 0.5) = 508.0$, and $\text{FPALOCY} = (1016/5) * (4 - 0.5) = 711.2$. These keywords are used by AWAIC to match a specific PRF from the input list to the pixel being processed.

- The $n \times n$ PRF images, characteristic to the predefined grid squares, are then supplied as an input list to AWAIC (parameter $-f4$). Note that only square grids are allowed, i.e., the number of input PRF images must be a perfect square: 1, 4, 9, 16, 25 etc. Multiple input PRFs are usually only needed if one is after accurate resolution enhancement (input parameter $-n > 1$). For simple co-addition ($-n = 1$), it is sufficient to use one (averaged) PRF for the entire array. The author has yet come across a detector whose PRF is highly variable to warrant using several or more PRFs for simple co-addition.
- The PRFs must be volume-normalized to unity. This is internally checked. The sum of all PRF values in an input PRF image must not differ from unity by more than $1.0\text{E-}06$. This tolerance is hard-coded in the *awaic.h* include file.
- The maximum linear mosaic dimension supported by the image projection libraries in AWAIC is 16° . However, one is likely to run out of memory first (depending on the output pixel scales chosen) before the necessary arrays are allocated. The reason for this maximum is that for sizes greater than this, the SIN and TAN projections (the most common types) will give pixel scale distortions of $>1\%$ and $>2\%$ respectively at the extremities relative to the mosaic center. The baseline specification for WISE is to have Atlas Image sizes no larger than ≈ 1 square degree, so this won't be a problem. The maximum linear size of 16° is hard-coded in the *awaic.h* include file.
- The internal *cell* to output mosaic pixel size ratio (command-line parameter $-pc$) must be expressible in the form $1/\text{integer}$, where *integer* = 1, 2, 3..., and lie within the range: $0.2 \leq pc \leq 1.0$. The lower limit is hard-coded in the *awaic.h* include file.
- The output mosaic-to-input image pixel size ratio (command-line parameter $-ps$) can be any fractional value and lie in the range: $0.1 \leq ps \leq 1.0$. The lower limit is hard-coded in the *awaic.h* include file.
- If the absolute output mosaic pixel scale parameter $-pa <\text{in arcsec}>$ is specified, it will over-ride the $-ps$ (mosaic-to-input pixel ratio) parameter. The input value for $-pa$ must satisfy: $0.1 * \text{minrawscale} \leq pa \leq \text{minrawscale}$, where $\text{minrawscale} = \min[\text{input_image_CDELT1}, \text{input_image_CDELT2}]$ and the *input_image_CDELTs* are pixel scales in the input raw images.

- The input pixel mask FITS images if specified, are expected to have a BITPIX=32 (i.e., 32-bit signed integer format). However, only the first 31 bits (excluding the sign bit) are used in processing. Masks with BITPIX=16 or 8 or even -32 (floating point) can still be stored. Only the integer part of the float will be stored for BITPIX=-32.
- The input fatal bit-string template specification: `-m <inp_fatalmask_bits>` allows one to flag pixels according to certain conditions/criteria. The bit definitions for WISE are outlined in the SIS document *WSDC D-1101* referenced in §1.3. The maximum value allowed is $2^{31} = 2147483647$. If this value is specified, then all pixels with integer values >0 (or $\geq 2^0$ – i.e., all flagging criteria) in the input masks will be omitted from the co-add.
- The area-overlap weighting interpolation method (command-line option `-if = 1`) can only be invoked when (i) generating a simple (`-n = 1`) co-add, **and** (ii) when *no rotation* of the input PRF during re-projection is desired (command line option `-rf = 0`). The interpolation method will default to the “nearest neighbor” method (`-if = 0`) if either rotation of the PRF is specified (`-rf = 1`), or, resolution enhancement is desired (`-n > 1`).
- It is recommended that the “nearest neighbor” method with rotation included (`-if = 0` and `-rf = 1`) be used when performing resolution enhancement.
- If “nearest neighbor” interpolation is used and the output mosaic pixel scale (from either `-pa` or `-ps` parameters) is *not* a fraction expressible as $(1/integer) \times \text{input image pixel scale}$ where $integer = 1, 2, 3, \dots$, then systematic patterns in the output depth-of-coverage map may result. These are normalized out of the intensity images since there is an implicit division by the coverage at every location (see formalism in §6.3). However, to *minimize* systematic variations in the coverage map when nearest neighbor interpolation is used, it is advised that: (i) a new output mosaic pixel scale be chosen that satisfies the $(1/integer) \times \text{input image pixel scale}$ criterion, and (ii) that the input PRFs are sampled to a pixel scale $\leq 0.25 \times \text{input image pixel scale}$. If you are adamant on using a specific output mosaic pixel scale, then the more robust (but slower) area-overlap weighting PRF-interpolation method (`-if 1`) can be used.



01,05	02,05	03,05	04,05	05,05
01,04	02,04	03,04	04,04	05,04
01,03	02,03	03,03	04,03	05,03
01,02	02,02	03,02	04,02	05,02
01,01	02,01	03,01	04,01	05,01

Figure 2: Example of a 5×5 partition of an input frame for defining FPA position-dependent PRFs with an indexing scheme for computing the FPALOCX and FPALOCY keywords (see §6.1).

6.2 Processing Phases

The main processing steps (on a broad level) are shown in Figure 3. The assumed (or recommended) pre-processing steps for the image inputs were described in §4. The first step is to check all inputs for consistency according to the assumptions outlined in §6.1, and then assign defaults to the optional input parameters. The program then sets up the WCS of an internal co-add *cell* grid based on the input parameters. This grid is not the same as the final co-add grid that represents the output image products. More details on this internal *cell* grid are given in §6.3.

The “AWAIC brain” is next and executes the main computational loop of the program. This is expanded in Figure 4 and further discussed below. The co-add and ancillary product arrays are constructed and incremented internally on the *cell* grids as pixels are read from each input frame. These are projected and undistorted if necessary by keying off specific FITS keywords. The projection and distortion correction occurs in one fell swoop, in the same transformation, and uses the ‘fast’ plane-to-plane projection library developed at the SSC (*libtwoplane*). The input frames are read and stored one at a time from disk to avoid memory overflow.

The final step involves down-sampling the *cell* co-add grid arrays if they were initially sampled on a finer scale than the desired output pixel scale. After re-sampling, these arrays are written to image files on disk in FITS format. The primary output products were described in §3.2.

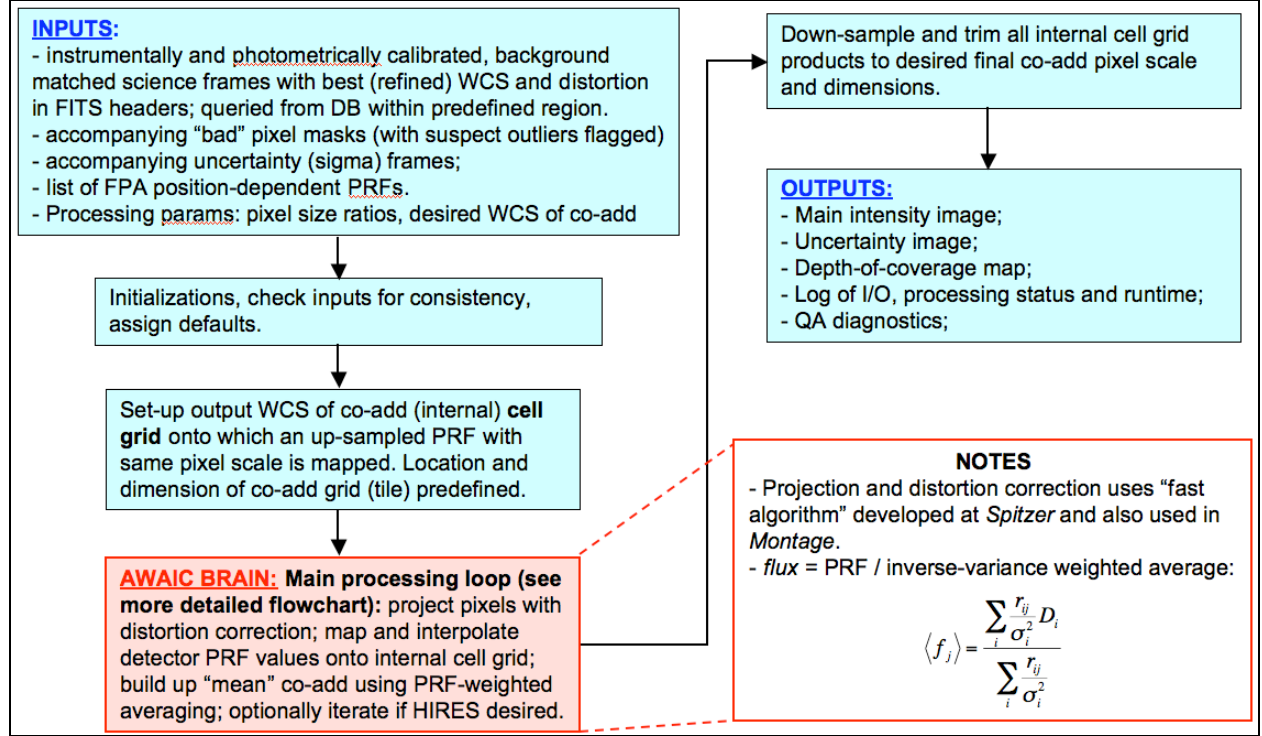


Figure 3: Top-level processing flow in AWAIC

6.3 Algorithm (the AWAIC brain)

We now outline the algorithm for simple image co-addition and *optional* resolution enhancement as implemented in AWAIC. The algorithm is based on the Maximum Correlation Method (MCM). This was previously implemented in IPAC's HIRES tool and still used to generate FRESCO image products from IRAS data. AWAIC includes generic functionality to handle images in standard FITS format, omit masked input pixels, allow non-isoplanatic PRFs, include prior-noise weighting, estimate uncertainties, and suppress ringing artifacts from the resolution enhancement.

MCM essentially incorporates classic Richardson-Lucy, but *first and foremost*, it is an image-reconstruction algorithm. MCM's goal is to build a model of the sky (truth) which reproduces and is "maximally consistent" with the observations via transfer by the PRF. The main steps of the MCM (as well as those leading to a simple co-add) are shown in Figure 4. We expand on these below.

- I. First, we begin by assuming a flat, nonzero model of the sky (the model co-add image):

$$f_j^n = \text{constant} \forall j \quad (\text{Eq. 4})$$

where the subscript j refers to a pixel in the model co-add image, and n refers to the iteration number. Eq. 4 only pertains to the very first iteration, $n = 1$. This starting model image is a first guess of the "truth" that we plan to gradually reconstruct and refine. Obviously it's a bad approximation, since it represents what we know without any measurements having been used yet, i.e., a featureless sky, but we'll fix it as you'll see.

- II.** Next, we use the detector PRF(s) to "observe" this model (flat) image. We predict the observed flux, F_i , in each detector pixel i from the model image by convolving it with the characteristic PRF centered at each detector pixel:

$$F_i^n = \sum_j r_{ij} f_j^n \quad (\text{Eq. 5})$$

r_{ij} is the response (PRF value) from detector pixel i at the location of model co-add pixel j . Note that the r_{ij} are volume-normalized to 1. Eq. 5 is in essence a "tensor inner product" of the model image and the PRF over which it spans, centered at the location of measurement pixel i in the model image-frame. It is not a convolution in the true sense of word since the convolution kernel r_{ij} may be non-isoplanatic (i.e., vary with location) on an array.

- III.** Correction factors are computed for each detector pixel i by dividing their observed flux, D_i , by that predicted from the model (Eq. 5):

$$C_i^n = \frac{D_i}{F_i^n} \quad (\text{Eq. 6})$$

- IV.** For each pixel j in the model co-add, all "contributing" correction factors (i.e., contributed by the overlapping PRFs r_{ij} of all neighboring detector pixels i) are averaged using response-weighted averaging:

$$\langle C_j^n \rangle = \frac{\sum_i \frac{r_{ij}}{\sigma_i^2} C_i^n}{\sum_i \frac{r_{ij}}{\sigma_i^2}} \quad (\text{Eq. 7})$$

The inclusion of inverse variance weighting ($1/\sigma_i^2$) is optional, but recommended if one believes in their measurement uncertainties σ_i . The baseline plan for WISE is to always include variance weighting when uncertainties are available.

- V. Finally, the original (or starting) model image pixels are multiplied by their respective averaged correction factors (Eq. 7) to obtain new estimates of co-add model fluxes:

$$f_j^{n+1} = f_j^n \langle C_j^n \rangle \quad (\text{Eq. 8})$$

If we are simply after a PRF-interpolated co-add/mosaic, we terminate the process here. The final co-add pixel fluxes are then given by $f_j^{n=2}$, where we started with a flat model image: $f_j^{n=1} = \text{constant}$ (Eq. 4).

If we desire resolution enhancement, the process is iterated by using the new and "better" model sky from Eq. 8 as input into Eq. 5. Thus, the model is continuously "re-observed" with the detector-PRFs to keep on producing a better model sky (Eq. 8). With patience, this eventually becomes the final co-add image with resolution that may exceed the diffraction limit of the optical system. The iteration stops when either the difference between the model pixel correction factors from successive iterations: $|C_j^n - C_j^{n-1}|$ becomes tiny (below some threshold), and/or the C_j^n values themselves are close to unity. When this is satisfied, the process has converged and the sky flux is said to have been resolved to an extent determined by the accuracy of the PRFs. So, subjecting a simple model image to the above simulation and feedback correction process "decorrelates" the model pixels. It is an algorithmic property of MCM that this decorrelation modifies the starting (flat) model image only to the extent necessary to make it reproduce the measurements to within the noise.

It's important to keep in mind that the very first iteration of MCM (which starts with a flat model at Eq. 4 and terminates at Eq. 8) yields the PRF-interpolated co-add/mosaic. This will be the default for all WISE Atlas Image and co-add products. Further iterations imply resolution enhancement.

In fact, steps I, II, III and IV above are not even needed when generating a simple PRF-interpolated co-add. This can be seen if we assume $f_j = 1 \forall j$ for the initial flat model image (Eq. 4). This implies $F_i = 1$ always (Eq. 5) since the PRF is volume-normalized to unity. Hence, $C_i = D_i$ in Eq. 6. Therefore, Eqs 7 and 8 can be combined to write an explicit expression for the flux in pixel j of a PRF-interpolated co-add:

$$f_j = \frac{\sum_i \frac{r_{ij}}{\sigma_i^2} D_i}{\sum_i \frac{r_{ij}}{\sigma_i^2}} \quad (\text{Eq. 9})$$

The processing in AWAIC is set up such that if a simple PRF-interpolated co-add is desired, the steps pertaining to MCM and resolution enhancement are bypassed and Eq. 9 is used instead.

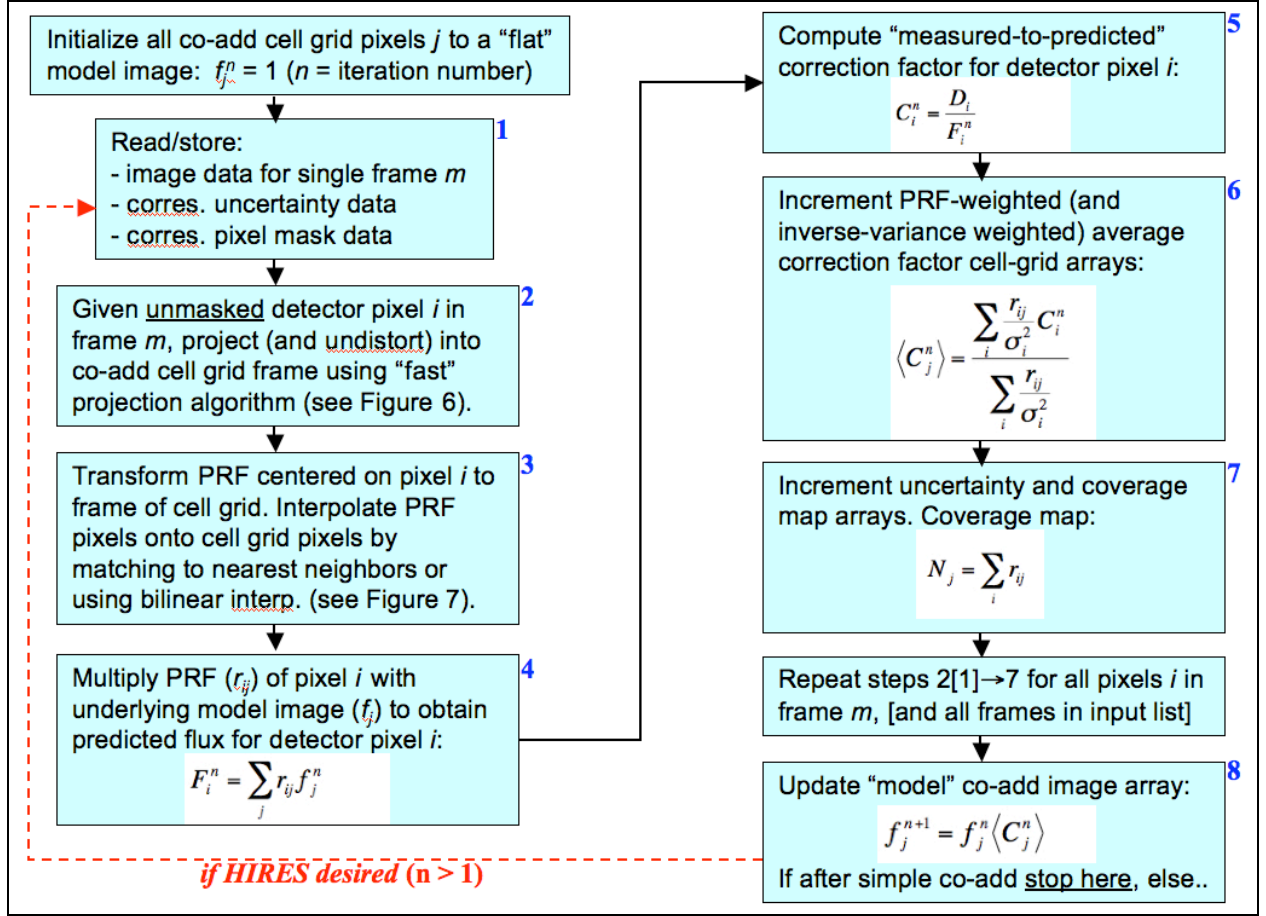


Figure 4: Main computational processing loop in AWAIC

Accompanying the primary co-add intensity image are a depth-of-coverage map, and an uncertainty image (box 7 in Figure 4). The depth-of-coverage map is an image of the same size as the co-add. This effectively indicates how many times a point on the sky (co-add) was visited by the PRF of a "good" detector (FPA) pixel, i.e., not rejected due to prior-masking, cosmic rays or other transients. The depth-of-coverage at a co-add pixel j is given by the sum of all overlapping (unit volume-normalized) PRF contributions at that location:

$$N_j = \sum_i r_{ij} \quad (\text{Eq. 10})$$

The uncertainty image represents a 1-sigma error estimate in the co-added signal at every pixel. Given an uncertainty σ_i for an input detector pixel i , the uncertainty in the response-weighted average (Eq. 9) at co-add pixel j is given by:

$$\sigma_j = \frac{\left[\sum_i \frac{r_{ij}^2}{\sigma_i^2} \right]^{1/2}}{\sum_i \frac{r_{ij}}{\sigma_i^2}} \quad (\text{Eq. 11})$$

These co-add uncertainties implicitly contain the result of the full error-propagation from all instrumental calibration steps upstream (§4.2). In other words, the detector measurement uncertainties σ_i are initiated using an error-model and progressively updated in processing. They are also appropriately re-scaled for consistency with the degree of repeatability over multiple frame exposures. It's also important to note that Eq. 11 is only applicable to the very first iteration of MCM, i.e., that generates the PRF-interpolated co-add (Eq. 9).

6.4 Interpolation onto the “cell-pixel” grid

The internal co-add *cell* grid mentioned in §6.2 is used to improve the accuracy of an interpolated PRF value at a co-add pixel location. This is required for Eqs 5 and 7, or 9 (for a simple co-add). An example of the various input and output pixel sizes used in AWAIC processing is shown in Figure 5. The desired output co-add pixel size is controlled by either the mosaic-to-input image pixel scale ratio (input parameter `–ps`), or, the absolute pixel scale in arcsec (input parameter `–pa`). The *internal working* cell pixel size is controlled by the mosaic *cell*-to-output mosaic pixel size ratio (input command-line parameter `–pc`). The *cell* pixel size can be selected according to the accuracy to which a PRF can be positioned and hence interpolated onto a co-add grid. Since the PRF is subject to thermal fluctuations in the optical system as well as pointing errors across multiple frame exposures, it does not make sense to have a cell-grid finer than the measured positional accuracy of the PRF.

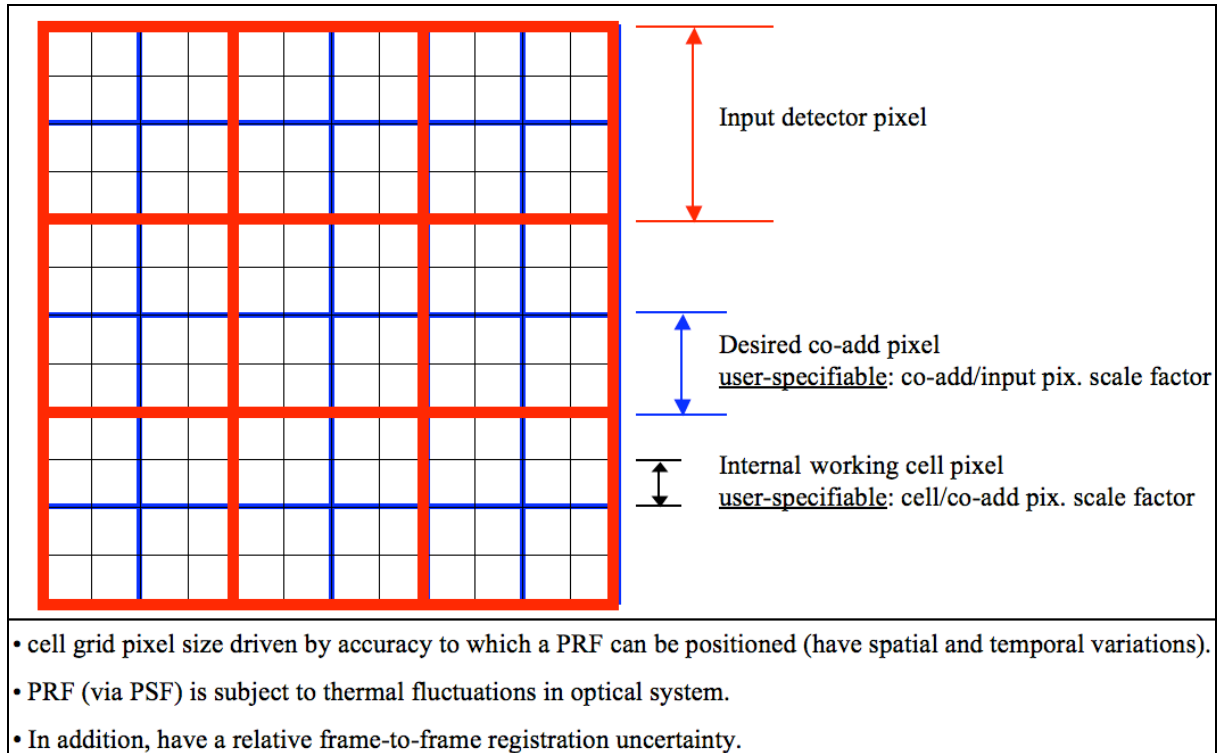


Figure 5: Input versus Co-add versus Cell-grid pixel definitions

Having an internal pixel grid that is sampled to a finer scale than the final co-add can significantly improve the accuracy of the interpolation, and this is crucial for optimum resolution enhancement. The requirement for interpolation accuracy for a simple (one-iteration) co-add is not as strict. A fine cell grid is most important for the *nearest neighbor* interpolation method, although the area-overlap weighting method will benefit too. Area-overlap weighting is advantageous when one has coarsely sampled PRFs. A coarsely sampled PRF (with presumably fewer pixels overall) can lead to significant improvements in runtime, however, to obtain good interpolation accuracy, it is advised that area-overlap weighting of PRF pixels onto the cell grid also be used. In AWAIC, area-overlap weighting can only be invoked when generating ‘simple’ *unHIRES*’d co-adds (see §6.1 for specific restrictions).

A schematic of a detector-PRF mapped onto the cell pixel grid is shown in Figure 6. The goal is to determine the value of the PRF at the location of a *cell* pixel j . Unless one specifies a small output co-add pixel size, the actual final co-add grid may be too coarse to perform a reliable interpolation. One can therefore make use of the internal *cell* grid to improve the interpolation accuracy. When the interpolation is complete, the program down-samples to the final desired co-add pixel size if a smaller internal co-add *cell* pixel size was indeed specified.

The detector PRF is transformed into the cell-grid frame by first projecting the center of a detector pixel using the standard WCS projection (with distortion correction if necessary), and then transforming the coordinates of the PRF pixels centered at this pixel into the *cell*-grid frame.

This latter transformation is accomplished using a local translational + rotational transformation as follows:

$$\begin{pmatrix} C_x \\ C_y \end{pmatrix} = \begin{pmatrix} D_x \\ D_y \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} P_x - P_x^0 \\ P_y - P_y^0 \end{pmatrix},$$

where (P_x, P_y) are PRF pixel coordinates and (P_x^0, P_y^0) center coordinates in the PRF frame, (D_x, D_y) are the detector pixel coordinates in the cell-grid frame, and the outputs (C_x, C_y) are the coordinates of an arbitrary PRF pixel in the cell-grid frame. The angle θ in the associated rotation matrix is effectively the rotation of the input image with respect to the co-add frame. When no rotation of the input PRFs are specified (i.e., input option `-rf 0`), we explicitly set $\theta = 0$. “Nearest-neighbor” interpolation entails rounding the output coordinates (C_x, C_y) to the nearest integer pixel. As mentioned above, if the input PRFs are too coarsely sampled, then systematic variations may result in the interpolated products (see §6.1 for details on how to circumvent this).

Under the area-overlap interpolation scheme, the contribution from an input PRF pixel (for a given detector pixel) is apportioned to the four neighboring pixels with which it overlaps according to overlap area. A schematic is shown in Figure 6.1. Given overlap areas a_{ijk} between a PRF pixel j (from detector pixel i) and four neighboring cell-grid pixels k of the *same size*, equations 9, 10, and 11 for the output cell pixel flux, depth-of-coverage, and uncertainty respectively need to be modified. These are replaced by:

$$f_k = \frac{\sum_i \sum_j^{N_{PRF}} \frac{a_{ijk} r_{ij} D_i}{\sigma_i^2}}{\sum_i \sum_j^{N_{PRF}} \frac{a_{ijk} r_{ij}}{\sigma_i^2}} \quad (\text{Eq. 9.1})$$

$$N_k = \sum_i \sum_j^{N_{PRF}} a_{ijk} r_{ij} \quad (\text{Eq. 10.1})$$

$$\sigma_k = \frac{\left[\sum_i \frac{1}{\sigma_i^2} \left(\sum_j^{N_{PRF}} a_{ijk} r_{ij} \right)^2 \right]^{1/2}}{\sum_i \sum_j^{N_{PRF}} \frac{a_{ijk} r_{ij}}{\sigma_i^2}} \quad (\text{Eq. 11.1})$$

This construction assumes that every PRF pixel j corresponding to the *same* detector pixel i has the *same* set of overlap areas $a_{ijk} = \{a_1, a_2, a_3, a_4\}_{ij}$ with respect to its four neighboring cells - see Figure 6.1. Different detector pixels will have different PRF-to-cell grid overlap areas due to

variable distortion in the focal plane. The overlap-areas are normalized such that for a cell pixel k overlapping with the PRF pixels of one detector pixel, their sum satisfies:

$$\sum_j^{N_{PRF}} a_{jk} = 1.$$

Due to programming complexity, Eq. 11.1 for the uncertainty was not implemented in its true form. Instead, we have assumed that the effective PRF value at cell pixel k is approximately equal to the PRF pixel value that's closest to that cell. In essence, this means $a_{jk} = \delta_{jk}$ where $\delta_{jk} = 1$ for $j =$ (or closest to) k , and $\delta_{jk} = 0$ for $j \neq k$. Eq. 11.1 then reduces to the more tractable Eq. 11 and the uncertainty becomes equivalent to that estimated using nearest neighbor interpolation. This approximation is fairly accurate if the input PRFs are well sampled (e.g., with pixel scale $\leq 0.25 \times$ input image pixel scale).

It's important to note that the PRF transformed in this manner (regardless of interpolation method) does not use the full *non-linear* WCS transformation from input image to co-add frame. For large mosaics ($>10^\circ$), there could be an inaccurate representation of the PRF flux distribution towards the mosaic edges. Also, the PRFs are not corrected for distortion (i.e., their non-isoplanatic nature) when projected.

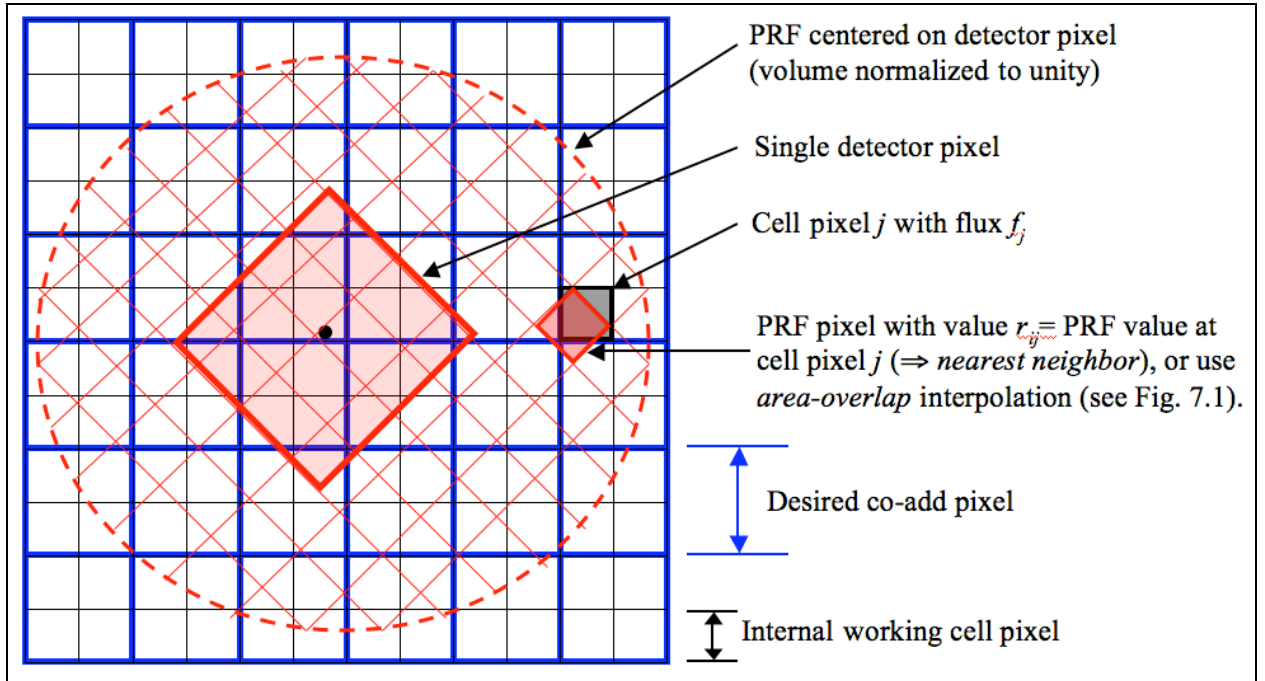


Figure 6: Detector-PRF placement and interpolation schematic

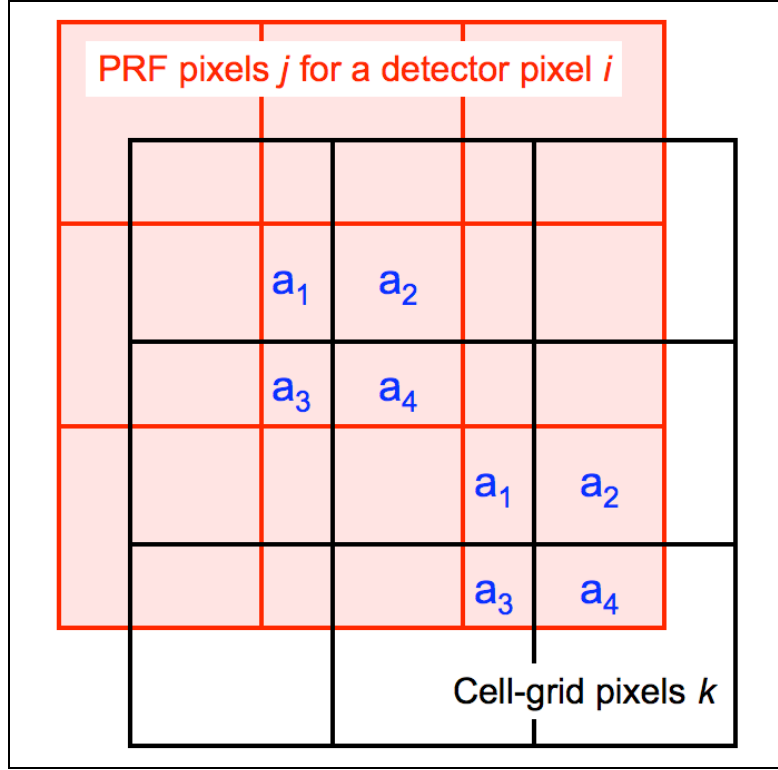


Figure 6.1: Schematic of PRF pixel-to-cell grid interpolation using area-overlap weighting. Four neighboring cell pixels are incremented in proportion to the PRF pixel overlap areas.

6.5 Pixel Overlap-Area Weighted Interpolation

An interpolation option which does not require knowledge of the PRF for co-addition is classic overlap-area weighting. This is similar to that implemented in the *Montage* and MOPEX tools and can be invoked with the “-sc 1” option. If specified, all inputs and computations pertaining to the PRF are ignored. This option can be used when one does not require the benefits of the PRF, e.g., if not interested in “matched filter” products for optimal point source detection (see §5).

In essence, this method involves projecting the four corners of an input frame pixel directly onto the final output co-add grid (with rotation included). Input/output pixel overlap areas are then computed using a textbook algorithm for the area of a polygon, and then used as weights when summing the contribution from many input detector pixels with signals D_i . The operation is analogous to that represented by Eq. 9 with the response function weights r_{ij} replaced by overlap-areas a_{ij} . The signal in co-add pixel j is given by:

$$f_j = \frac{\sum_i \frac{a_{ij}}{\sigma_i^2} D_i}{\sum_i \frac{a_{ij}}{\sigma_i^2}}. \quad (\text{Eq. 9.2})$$

Corresponding coverage and uncertainty co-adds are also computed, analogous to Eqs 10 and 11.

An ancillary (and optional) product that can only be generated under the overlap-area weighting option (`-sc 1`) is an image of the standard-deviation of the input pixel stacks divided by the square-root of the depth-of-coverage at each location. This represents an alternative measure of the uncertainty in a co-add pixel, and is only generated if the `-o4 <outfile>` is supplied. In general, the variance (dispersion) in a pixel stack at location j can be written:

$$\text{var}_j = \frac{N_j}{N_j - 1} \left[\langle D_i^2 \rangle_j - \langle D_i \rangle_j^2 \right],$$

where N_j is the depth-of-coverage (effective number of input pixels) and the pre-factor $N_j / (N_j - 1)$ is the correction to convert the sample variance to an unbiased estimate of the population variance. The angled brackets denote area-weighted averages of all the input detector pixel signals overlapping with co-add pixel j . The standard-deviation uncertainty in the co-add signal is then given by

$$\sigma_{SDj} = \sqrt{\frac{\text{var}_j}{N_j}}.$$

Using the above expressions, where $\langle D_i \rangle_j \equiv f_j$ is given by Eq. 9.2, the standard-deviation uncertainty can be written:

$$\sigma_{SDj} = \frac{1}{\sqrt{N_j - 1}} \left[\frac{\sum_i \frac{a_{ij}}{\sigma_i^2} D_i^2}{\sum_i \frac{a_{ij}}{\sigma_i^2}} - f_j^2 \right]^{1/2}.$$

This is only computed for pixels where $N_j > 1$, otherwise $\sigma_{SDj} = 0$ is written to the output image.

Note that σ_{SDj} is not appropriate for PRF-interpolated co-adds (i.e., the first iteration of MCM with `-n = 1`). This is due to a detector pixel flux being spread over a region in the co-add grid spanning the domain of the input PRF. This spreading leads to an artificial dispersion in the stack at any co-add pixel location. E.g., even a single flat image will have var_j non-zero everywhere and will represent a meaningless measure. Only after many MCM iterations, i.e., until the process has converged to yield the best resolution possible will the σ_{SDj} measure start to make sense. It then represents the dispersion within local pixels, and large values indicate an inconsistency between the input detector measurements (e.g., outliers). Estimates of σ_{SDj} for the generic MCM - HIRES case will be implemented in future.

For now, σ_{SDJ} is only implemented for simple overlap-area interpolated co-adds. In general, this measure is expected to be larger than the uncertainty estimated and propagated from a prior error model for the input pixels (e.g., Eq. 11). This is because many more sources of error can contribute to the dispersion in a pixel stack, e.g., under-sampling of PSFs (even close to critical), pointing and registration errors, and frame-to-frame systematics due to inaccurate instrumental calibration, and/or unforeseen transients.

6.6 Command-line Usage Examples

The following example reads in a list of image FITS file names, corresponding masks, uncertainty images and PRF images. A mosaic of dimensions $0.785^\circ \times 0.785^\circ$ centered at RA, Dec = 220° , 80° , orientation of 0 (with respect to north) and output pixel scale (`-pa`) of 1 arcsec is generated. An internal cell-grid pixel size (`-pc`) of 0.5*output co-add pixel scale is used with *nearest neighbor* interpolation (`-if 0`) for the PRF. There is no rotation of the PRF (`-rf 0`) when projected. To use the *more accurate* (but slower) area-overlap interpolation option, set “`-if 1`”. The value of `-pc` is based on the positional accuracy of the detector’s measured PRF and must equal the ratio of input PRF to output co-add pixel scale to a difference tolerance of `-ct 0.0001` arcsec. The output co-add pixel values are scaled according to pixel size (`-sf 1`). In other words, the input image pixel values are in DN (counts) and not in surface brightness units. No HIRES’ing is invoked (`-n 1`). Also, the fatal mask bit-string template (`-m`) is set to 132096 ($=2^{10} + 2^{17}$). This means that all pixels with bits 10 and 17 turned on in the input masks will be omitted from the co-add. Coverage and uncertainty mosaics are also generated. The verbose switch is also set (`-v`) which means details of processing are written to standard output.

```
% awaic -f1 ImageList.txt -f2 MaskList.txt -f3 UncertList.txt -
f4 PRFList.txt -X 0.785 -Y 0.785 -R 220.0 -D 80.0 -C 0.0 -m
132096 -pa 1.0 -pc 0.5 -sf 1 -n 1 -rf 0 -ct 0.0001 -if 0 -o1
mosaic.fits -o2 mosaic_cov.fits -o3 mosaic_unc.fits -v
```

The following example is similar to the above except that now, no input mask or uncertainty images are available. A simple (`-n 1`) co-add and coverage map with 1 arcsec square pixels are generated. The `-C`, `-pc`, `-sf`, `-rf`, `-ct`, `-if`, and `-n` parameters as specified above are also internal defaults and don’t need to be included. If a HIRES’d co-add is desired, then include `-n <iter>` where *iter* is the desired number of iterations.

```
% awaic -f1 ImageList.txt -f4 PRFList.txt -X 0.785 -Y 0.785 -R
220.0 -D 80.0 -pa 1.0 -o1 mosaic.fits -o2 mosaic_cov.fits -v
```

If one is after a quick co-add made using the conventional input-to-output area-overlap weighting method (i.e., similar to that used by the MOPEX and *Montage* tools), then the user need only specify “`-sc 1`” on the command line. All inputs/computations pertaining to the PRF are ignored. Here’s an example synopsis for simple area-overlap weighting, where masked pixels are omitted from the co-add:

```
% awaic -f1 ImageList.txt -f2 MaskList.txt -f3 UncertList.txt -X
0.785 -Y 0.785 -R 220.0 -D 80.0 -C 74.0 -m 132096 -pa 1.2 -sc 1
-o1 mosaic.fits -o2 mosaic_cov.fits -o3 mosaic_unc.fits -o4
mosaic_std.fits -v
```

7 EXAMPLES AND TESTING

A number of co-adds (including HIRES'd versions too) from simulated and real image data appear in Section IV of the AWAIC website:

<http://web.ipac.caltech.edu/staff/fmasci/home/wise/awaic.html#EG>

Here you will find the following test cases:

- A simple simulation consisting of eight interleaved image frames each matching the WISE FOV. Each contains point sources simulated using an approximate WISE band-1 PRF, single-spike radiation hits, Poisson noise, and assumed bad-pixels flagged in masks. The goal here was to assess accuracy, speed and how close the MCM-HIRES algorithm could reconstruct the input truth.
- Real data from the IRAC detector on *Spitzer*. This example is a map of NGC2403 and consists of 68 input image frames. Comparisons with the *Montage* tool are also made.
- Observations of the North Ecliptic Pole (NEP) – the Northern “WISE Touchstone Field” with IRAC and MIPS-24 μ m. This field will be observed on nearly every orbit. Comparisons with the MOPEX tool are also made. Results of a study on flux conservation from input frame to co-add frame and vice-versa are also presented.
- Case study of a bright star in the NEP mosaic above. This explores the impact of masked (missing) pixels in AWAIC by examining what happens to the cores of bright stars that have been masked due to saturation.
- Also in http://web.ipac.caltech.edu/staff/fmasci/home/wise/sep_mosaics.html, you will see example mosaics generated using of the full suite of CO-ADD modules (*Bmatch*, *AWOD* and *AWAIC*) on 2MASS, IRAC and MIPS observations of the South Ecliptic Pole (SEP).

8 QUANTITATIVE ANALYSIS

This section uses AWAIC to explore the following:

1. How does the photometric noise in a co-add pixel scale with the number of frame overlaps (depth-of-coverage)?

2. How does the uncertainty in the total flux measured in an aperture on a co-add scale with aperture size and depth-of-coverage?
3. On a related note, what is the effect and magnitude of spatial correlations on aperture fluxes, and is flux conserved within measurement error?

We have addressed these questions by simulating 1000 randomly dithered, overlapping image frames. All dithers were made to overlap within a central region of a co-add grid spanning $\sim 600 \times 600$ pixels at 1.375 arcsec/pixel. Before making the random frame cutouts, a constant background of 1000 counts per pixel was assumed. To assist in our aperture photometry analysis, we also added a single "truth" point source of 500 counts at the center of the co-add grid. We then convolved with a Gaussian kernel of $\sigma = 2.75$ arcsec (=one input frame pixel) truncated at 3-sigma. Pure Poisson noise was added to each frame cutout by sampling from a Normal distribution with variance equal to the pixel (mean) value. Sets of 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512 frames were then collected and co-added with AWAIC to make 10 separate co-adds.

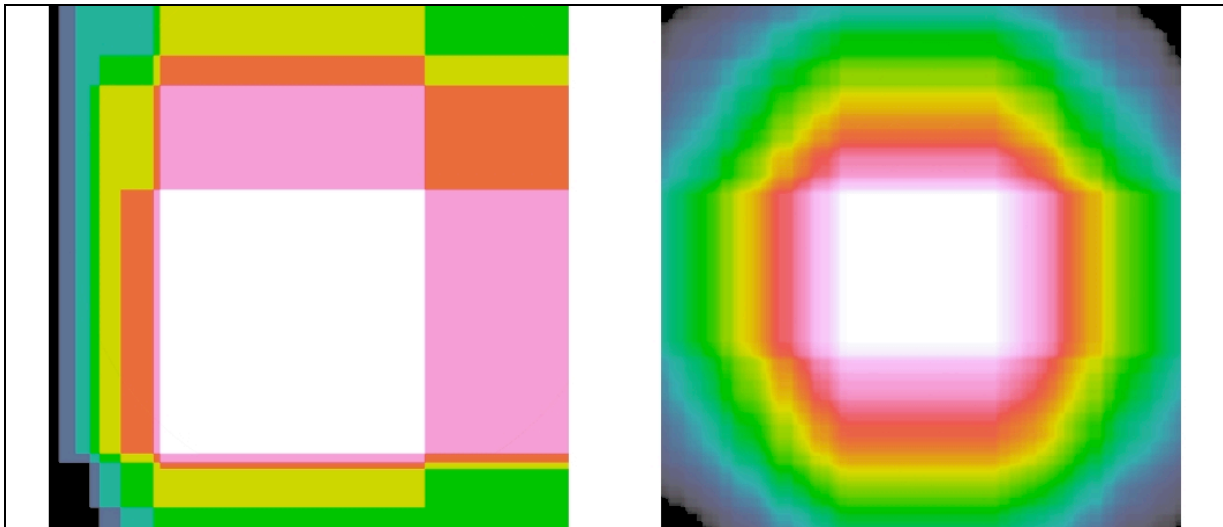


Figure 7: depth-of-coverage maps for two simulated co-adds. Left: 16-frames; Right: 256 frames. The central (white) portions contain the maximum depths.

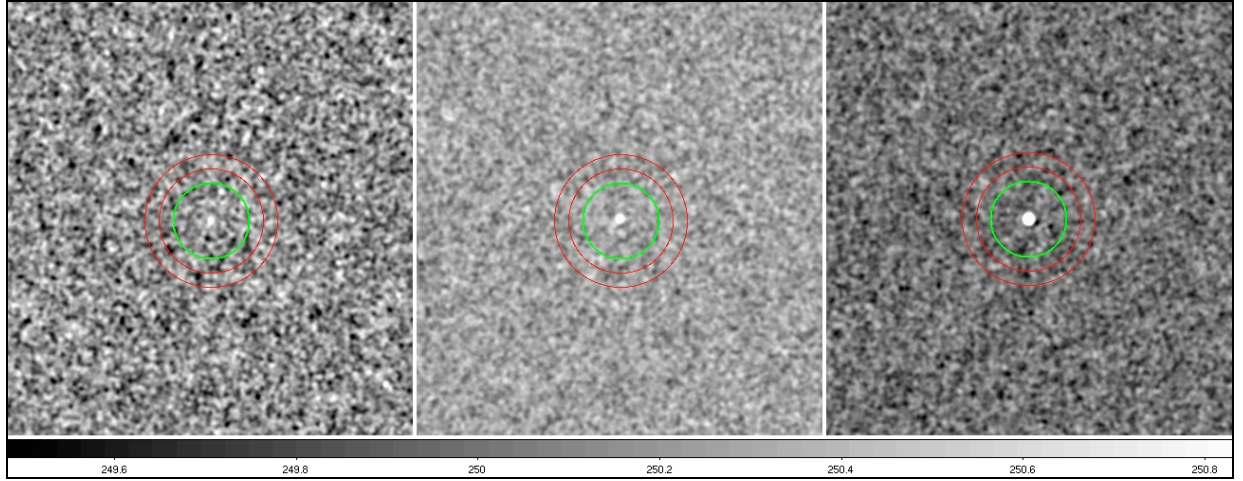


Figure 8: simulated co-adds from AWAIC corresponding to maximum central coverage depths of 1 (left, i.e., a single interpolated frame), 8 (middle), and 256 (right). The green circle is a test aperture for our aperture photometry and the red annulus is for the background estimation. These are centered on our "known" simulated point source and is barely visible on the "1-frame co-add".

8.1 Co-add Pixel Noise

For a *constant* background level, constant Poisson noise derived there from, and an isoplanatic PRF, it is straightforward to show from Eq. 11 that the uncertainty in a single co-add pixel j reduces to:

$$\sigma_j = \frac{\sigma_i}{n_{pix} \sqrt{N_f N_{npix}}} \quad (\text{Eq. 12})$$

where :

σ_i = measurement uncertainty for detector pixel i

n_{pix} = number of output (co - add) pixels per detector pixel

N_f = number of overlapping frames in co - add stack

$$N_{npix} = \frac{\left[\sum_i r_{ij} \right]^2}{\sum_i r_{ij}^2} \equiv \text{number of "noise pixels" (a characteristic of the PRF)}$$

For the PRF used in these simulations, the number of "noise pixels" is $N_{npix} \sim 12.46$ in input frame pixel units. The number of co-add pixels per input frame pixel is $n_{pix} = 4$. The uncertainty assumed per input pixel (in the background) is $\sigma_i = \sqrt{1000} \sim 31.62$. Figure 9 compares the predictions of Eq. 12 with measurements of the pixel RMS off the co-adds, and those computed

by AWAIC via full PRF-weighted averaging (Eq. 11). The noise per co-add pixel scales as expected: $\sim 1/\sqrt{N_f}$, i.e., as given by Eq. 12.

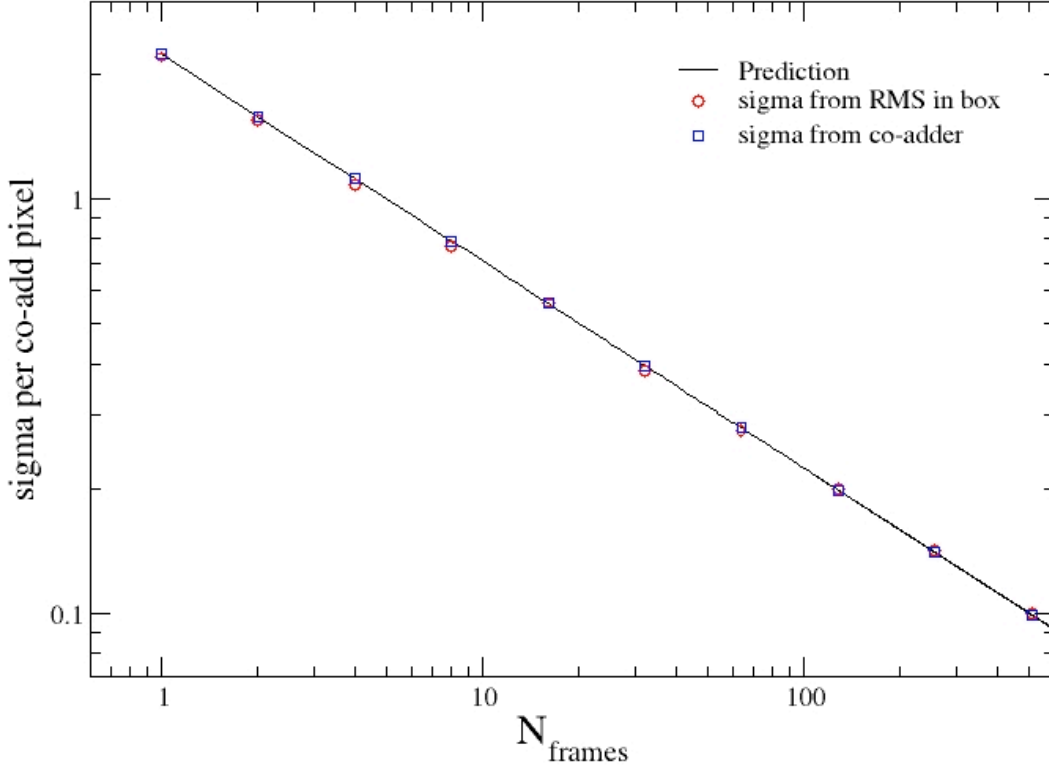


Figure 9: Co-add pixel 1-sigma uncertainty as a function of the number of frame-overlaps (1 to 512 frames). The solid line is the prediction given our simulation inputs (Eq. 12). The squares are the pixel uncertainties computed by AWAIC (Eq. 11) and are constant over space since we assumed a uniform background. The circles correspond to the RMS of pixel fluctuations in a box placed on the background of each co-add. The noise per co-add pixel scales as expected: $\sim 1/\sqrt{N_f}$ (Eq. 12). Since this is a log-log plot, the slope is -0.5.

8.2 Aperture Photometry and Correlated Noise

We performed aperture photometry on our simulated point source in all 10 co-adds. Three of these co-adds with overlaid apertures are shown in Fig. 9. We assumed an aperture radius of 10 co-add pixels throughout. This corresponds to $\sim 2 \times FWHM$ of the PRF. The background annuli were chosen to have approximately the same number of pixels as the source aperture. The total count in an aperture containing N_A pixels is the sum of all pixel counts:

$$F_{tot} = \sum_j^{N_A} f_j \quad (\text{Eq. 13})$$

The variance in the total count can be written in terms of the variance in pixel j , and the covariance between any two pixels (j, k) within the aperture:

$$\sigma_{F_{tot}}^2 = \sum_j^{N_A} \sigma_j^2 + 2 \sum_j^{N_A} \sum_k^{N_A} \text{cov}(j, k) \quad (\text{Eq. 14})$$

where :

$j > k$ and

$$(x_j - x_k)^2 + (y_j - y_k)^2 \leq D_{PRF}^2$$

The covariance or “correlated noise” is a consequence of a detector pixel PRF spanning a finite region over space. A noise fluctuation in a detector pixel affects all the co-add pixels in the PRF's domain after interpolation. The maximum range over which co-add pixels can be “coupled”, i.e., the “correlation length” D_{PRF} , is determined by the maximum linear extent of the PRF. The PRF used in these simulations spanned ~ 16 co-add pixels. Thus, the last condition in Eq. 14 should be apparent: only co-add pixels over the span of the PRF are coupled.

We shall now simplify Eq. 14 and express the covariance in terms of known (measurable) quantities. In general, the covariance between any two co-add pixels with uncertain (stochastic) fluxes (f_j, f_k) can be written:

$$\begin{aligned} \text{cov}(j, k) &= \left\langle (f_j - \langle f_j \rangle)(f_k - \langle f_k \rangle) \right\rangle \\ &= \langle f_j f_k \rangle - \langle f_j \rangle \langle f_k \rangle, \end{aligned} \quad (\text{Eq. 15})$$

where the angled brackets denote expectation value. Given that the main source of stochastic error is from the detector pixels themselves, let's assume that a detector pixel has a true flux D_i and a measurement error ε_i . Using Eq. 9, the inferred co-add flux in pixels j and k will be respectively:

$$\begin{aligned} f_j &= \frac{\sum_i \frac{r_{ij}}{\sigma_i^2} (D_i + \varepsilon_i)}{\sum_i \frac{r_{ij}}{\sigma_i^2}}, \\ f_k &= \frac{\sum_i \frac{r_{ik}}{\sigma_i^2} (D_i + \varepsilon_i)}{\sum_i \frac{r_{ik}}{\sigma_i^2}}. \end{aligned} \quad (\text{Eq. 16})$$

Substituting the expressions from Eq. 16 into Eq. 15, and evaluating the expectation values with the assumption:

$$\langle \varepsilon_m \varepsilon_n \rangle = \delta_{mn} \sigma_i^2,$$

where

$$\delta_{mn} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n, \end{cases}$$

i.e., the detector pixels in a frame (as well as across frames) are *uncorrelated*, the covariance can be written:

$$\text{cov}(j, k) = \frac{\sigma_i^2}{n_{pix}^2 N_f C_{jk}} \quad (\text{Eq. 17})$$

where :

$$C_{jk} = \frac{\left(\sum_i r_{ij} \right) \left(\sum_i r_{ik} \right)}{\sum_i r_{ij} r_{ik}}$$

Analogous to the “number of noise pixels” N_{npix} in Eq. 12, the quantity C_{jk} can be defined as the “number of *correlated* noise pixels”, and is also a characteristic of the PRF. The total variance in Eq. 14 can be written as the sum of an uncorrelated and a correlated term:

$$\begin{aligned} \sigma_{Ftot}^2 &= V_{unc} + V_{corr} \\ &= V_{unc} \left(1 + \frac{V_{corr}}{V_{unc}} \right) \end{aligned}$$

Combining Eqs 12, 14 and 17, the total variance in an aperture of N_A pixels can be written:

$$\sigma_{Ftot}^2 \approx N_A \sigma_j^2 \left[1 + 2 \frac{N_{npix} K_{N_A}}{N_A} \right] \quad (\text{Eq. 18})$$

where :

$$K_{N_A} = \sum_j^{N_A} \sum_k^{N_A} \frac{1}{C_{jk}}$$

Eq. 18 is an approximation because it assumes all co-add pixels within the aperture have the same variance (but see below for a more accurate expression). This applies under the assumption of a *constant* background and Poisson noise derived there from (e.g., analogous to Eq. 12 for a

single pixel). Using Eq. 12 to simplify Eq. 18, the uncertainty in the total counts within an aperture can be written:

$$\sigma_{F_{tot}} \approx \sqrt{\frac{N_A}{N_f N_{npix}} \frac{\sigma_i}{n_{pix}} F_{corr}(N_A)} \quad (\text{Eq. 19})$$

where :

$$F_{corr}(N_A) = \left[1 + 2 \frac{N_{npix} K_{N_A}}{N_A} \right]^{1/2}$$

$F_{corr}(N_A)$ is a correction factor ≥ 1 that accounts for an increase in the uncertainty due to correlated noise between co-add pixels in the aperture. This factor depends only on the PRF properties and the number of pixels in the aperture, N_A . This factor can be computed numerically for different detector PRFs and a range of aperture sizes. A user performing aperture photometry can then look up the appropriate value to apply from either a table or graph.

For a more accurate measure of the uncertainty in total counts in an aperture when one has a non-isoplanatic PRF, a non-uniform background, or, when there is much variation in pixel values (and Poisson noise) as one always has with a real source, the following expression is more appropriate:

$$\sigma_{F_{tot}} \equiv \left[\sum_j^{N_A} \sigma_j^2 \right]^{1/2} F_{corr}(N_A) \quad (\text{Eq. 20})$$

But note, this is still an approximation only with regards to how the correlation correction factor $F_{corr}(N_A)$ was derived. It was derived under the assumptions of (i) a *constant* background Poisson noise across all multiple frame exposures contributing to a co-add, and (ii) a single isoplanatic PRF. The σ_j in Eq. 20 are the actual co-add pixel uncertainties computed by AWAIC (Eq. 11). In general, if one wants to compute the uncertainty in total counts or flux in an aperture, they would take the square of the 1-sigma uncertainty image from AWAIC and sum the pixel variances in that aperture. They would then take the square root of this sum (as in Eq. 20), and multiply by the appropriate correction factor $F_{corr}(N_A)$ to account for correlated noise.

In general, the equations for source flux F_{src} and its variance from aperture photometry can be written:

$$\begin{aligned} F_{src} &= F_{obj} - N_A F_{Bck/pix} \\ \sigma_{F_{src}}^2 &= \sigma_{F_{obj}}^2 + N_A^2 \sigma_{F_{Bck/pix}}^2 \end{aligned} \quad (\text{Eq. 21})$$

where F_{obj} is the *total* count in the source aperture and $F_{Bck/pix}$ is the estimated background per pixel. Here the background is assumed to be uniform (slowly varying) in the vicinity of the target

source. We advise that the distance between of the inner boundary of the background annulus and the boundary of the source aperture be greater than the PRF correlation length in order to avoid possible correlations between noise fluctuations in the background and source flux. If the background annulus were too close to the source aperture (which you want to avoid for other reasons), we would need to add a covariance term to Eq. 21.

One can estimate the background per pixel in an annulus containing N_B pixels using either the mean:

$$\begin{aligned} F_{Bck / pix} &= \langle F_{Bck / pix} \rangle = \frac{1}{N_B} \sum_j^{N_B} F_{Bck,j} \\ \sigma_{FBck / pix}^2 &= \left[\frac{F_{corr}(N_B)}{N_B} \right]^2 \sum_j^{N_B} \sigma_{Bck,j}^2 \end{aligned} \quad (\text{Eq. 22})$$

or, via a more robust measure (robust against outliers), the median:

$$\begin{aligned} F_{Bck / pix} &= \text{median}(F_{Bck,j}) \\ \sigma_{FBck / pix}^2 &\approx 1.57 \left[\frac{F_{corr}(N_B)}{N_B} \right]^2 \sum_j^{N_B} \sigma_{Bck,j}^2 \end{aligned} \quad (\text{Eq. 23})$$

One can perform outlier filtering prior to computing these quantities, particularly if one decides to use the mean (Eq. 22). Eq. 20 was used to write the variance expressions in Eqns 22 and 23. It's important to note that the median is more uncertain than the mean when estimating the standard error with respect to the population. The factor of 1.57 ($\pi/2$ to be exact) accounts for this under the assumption that the background pixel values are close to Normally distributed. This is almost always ensured if N_B is large.

Figure 10 shows the result of our aperture photometry as a function of the number of frame-overlaps in a co-add. Eqns 20 - 22 were used to estimate fluxes and uncertainties. The 1-sigma error bars account for correlated noise in the source aperture as well as the background annulus. The correlation correction factors, $F_{corr}(N_A)$ and $F_{corr}(N_B)$, were first computed numerically from the PRF. With $N_A \sim N_B$, we found values of ~ 6 for both factors. These were then verified by placing 500 apertures at random within co-add regions with uniform coverage and computing the RMS of their total fluxes. This effectively gave the total uncertainty sought for, but, when divided by the uncorrelated RSS'd pixel sigmas (e.g., sum term in Eq. 20), correction factors of 6 ± 0.3 were found, in agreement with our numerical estimate.

In the end, ignoring correlated noise would have lead us to underestimate source-flux errors and hence overestimate signal-to-noise ratios by a factor of ~ 6 . This example shows the danger of ignoring correlated errors when determining photometric accuracy off a co-add made using an

interpolation kernel spanning many pixels. This applies to both aperture and profile fitting methods when performed directly off a co-add.

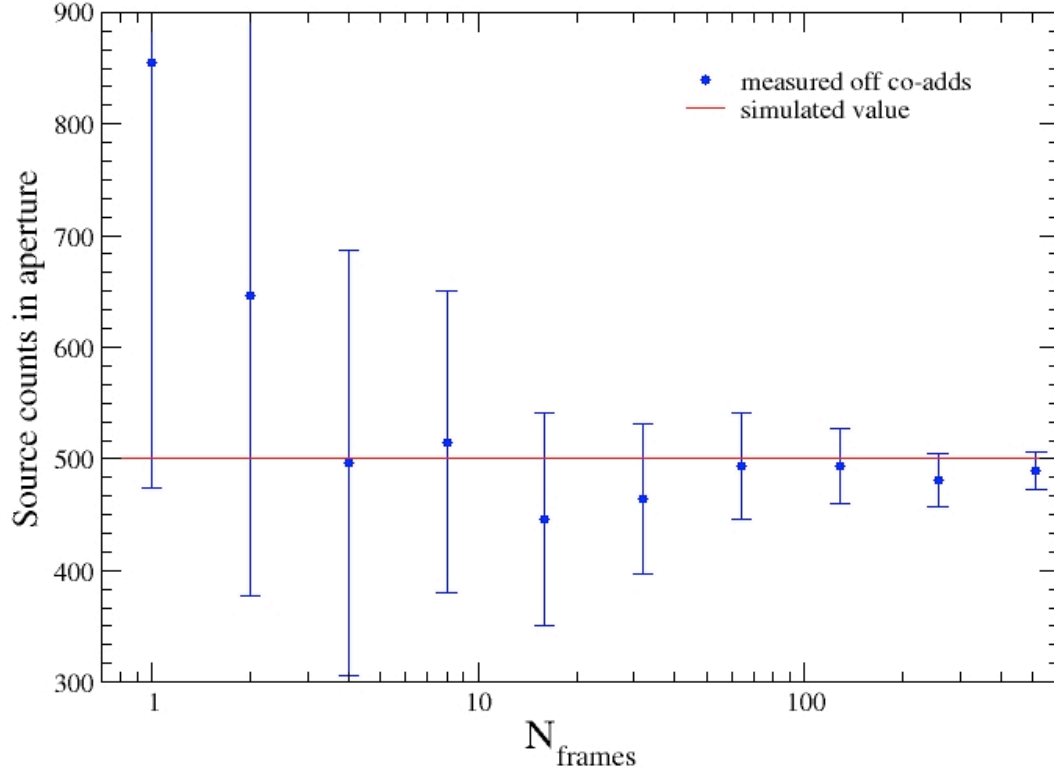


Figure 10: Total background-subtracted counts ("flux") in an aperture centered on our point source versus number of frame-overlaps in co-add. The solid horizontal line is the "truth". Overall, the measured counts are repeatable *to within measurement error* across all co-add depths. The error bars are all 1-sigma and account for spatial correlations between pixels (see §8.2).

9 WORK IN PROGRESS AND PLANS

Here's a summary of work in progress and future upgrades for both AWAIC, and the CO-ADD subsystem in general.

- Software to compute QA diagnostics and metrics (e.g., which frame pixels were omitted from co-addition and for what reason; noise characterization; coverage distributions etc). Frame stack statistics and χ^2 sanity checks.

- Look-up-table (and/or graph) of correction factors versus aperture size to account for correlated noise in co-adds. This is information for the Explanatory Supplement.
- *Low priority*: improve HIRES functionality (for offline operations): MCM convergence and χ^2 criteria; flux-bias to reduce ringing artifacts; correction factor variance (CFV) and computation of uncertainties.

10 REFERENCES

- 2MASS Explanatory Supplement (Co-addition Section):
http://www.ipac.caltech.edu/2mass/releases/allsky/doc/sec4_3.html
- Aumann, H.H., Fowler, J.W., and Melnyk, M., 1990, *A Maximum Correlation Method for Image Construction of IRAS Survey Data*, *Astronomical Journal*, **99**, pg. 1674
- Fowler, J.W., 2005, *MCM Reconstruction of Deep Impact Images II: Conclusions and Recommendations*, private communication.
- Fowler, J.W., and Aumann, H.H., 1994, *HiRes and Beyond*, in *Science with High Spatial Resolution Far-Infrared Data*, eds. S. Terebey and J. Mazzarella, (Pasadena, JPL), pg. 1.
- FRESCO Processing Overview: http://irsa.ipac.caltech.edu/IRASdocs/fresco_proc.html
- HIRES Overview: http://irsa.ipac.caltech.edu/IRASdocs/hires_over.html
- Montage - An Astronomical Image Mosaic Engine: <http://montage.ipac.caltech.edu/>
- MOPEX - MOsaicker and Point source Extractor:
<http://ssc.spitzer.caltech.edu/postbcd/mopex.html>
- Wells, B.S., 1997, *Far infrared to radio continuum correlation for nearby star forming galaxies*. Chapter 4: IRAS and the HIRES Simulator.

Acknowledgments

The author is indebted to Gene Kopan, John Fowler and Ken Marsh for illuminating and philosophical discussions. The author also thanks Roc Cutri, Tim Conrow and Tom Jarrett for advice and direction.