ASTRONOMICAL IMAGE SUBTRACTION BY CROSS-CONVOLUTION

Fang Yuan & Carl W. Akerlof

April 10, 2008

Software Installation and Execution

To execute the image subtraction code described in our ApJ paper, you will primarily need the IDL data visualization & analysis platform currently supported by the ITT Corporation. The second prerequisite is the IDL astronomy library that is freely available at http://idlastro.gsfc.nasa.gov/. All the rest of the required code is available in this archive stored in the University of Michigan Library Deep Blue online digital repository. These routines are listed below:

xconvolve_cross_convolve	xconvolve_extend_grid	xconvolve_get_sky
xconvolve_make_kernels	xconvolve_make_mask	xconvolve_subtract_images
	xconvolve_warp_image	

Cross-Convolution IDL functions & procedures

xconvolve_regress_matrix

External C subroutine

binary_search	close_match_radec	iqd
xconvolve_splie2	xconvolve_splin2	

Third-party IDL functions & procedures

The routines, *binary_search* and *close_match_radec* were written by Dave Johnston while he was at the University of Michigan. The routines *xconvolve_splie2* and *xconvolve_splin2* were derived from the codes for *splie2* and *splin2* obtained from the Web site, http://www.astro.washington.edu/deutsch/idl/. Permission to include these routines was kindly granted by their author, Eric W. Deutsch. The simplest procedure is to embed all of these routines in a common subdirectory, paralleling the storage of the NASA/Goddard IDL astronomy library. This will be platform and installation dependent. Path linkage should be accomplished in a similar fashion to the equivalent procedure for the NASA IDL package. The external C subroutine, *xconvolve_regress_matrix*, must be compiled into a sharable library (UNIX), or DLL (Windows). This should be taken care of automatically within the subroutine, *xconvolve_cross_convolve*, although we still have some compatibility issues with the Windows platform. *xconvolve_make_sharelib* has been provided to create a shareable library or dynamic link library so that the process of C-code compilation can be avoided at run time. Beware that depending on the installation, the appropriate directory for storing the shareable library/DLL for the C routine may not be accorded with write permission for all users. This makes initial compilation a desirable option. **Note: we are currently working on some compatibility**

problems associated with Windows XP and Visual Studio that are causing problems for MAKE_DLL and C code compilations. We hope to resolve these soon.

To test these procedures, we have provided an IDL test program, *sample_subtract.pro*. This code should be edited to reflect the syntax of your particular computing environment. Four data files have been included for input:

070802_sks1650+2342-164942+235329_3b007_c.fit
070802_sks1650+2342-164942+235329_3b007_cobj.fit
070820_sks1650+2342-164942+235329_3b003_c.fit
070820_sks1650+2342-164942+235329_3b003_cobj.fit

The basic flattened image data is supplied by *_c.fit* files. The cross-convolution code expects to find the parameter, SATCNTS, the image pixel saturation level, in the FITS file header to identify and reject saturated pixels. The object list files, *_cobj.fit*, should be obtained via initial processing with SExtractor, followed by calibration to the USNO-A catalog to refine astrometry and photometry. The first FITS extension of these files contains an array of structures. Useful tags are *ra*, *dec*, *x*, *y*, *m* (magnitude of object), *fwhm* (in pixels), and *flags* (≤ 2 for good objects). The second FITS extension contains a structure storing calibration information. The useful tags are *fwhm* (a typical fwhm of the image) and *zp_offset* (magnitude zero point to convert the magnitudes in the first extension to counts). These lists are used in *xconvolve_warp_image()* and *xconvolve_make_mask()*. Exactly which tags are required depend on choice of functions. For example, in a relatively crowded field where reliable counts for all objects are not easily obtained, the user may choose to scale the new image so that the total flux within the masked area matches that of the reference. In this case, the *scalenew* keyword will not be set in *xconvolve_make_mask()*; rather the user should perform the scaling in a separate step after a mask is produced.

The output of *sample_subtract* is a subtracted image file, *test.fit*, which should be compared with *sample_test.fit*.

This documentation is admittedly rather brief. We would appreciate your comments so that we can update this text to make the procedure more transparent for all potential users. We can be mostly easily reached at our E-mail addresses, `yuanfang@umich.edu` and `cakerlof@umich.edu`.

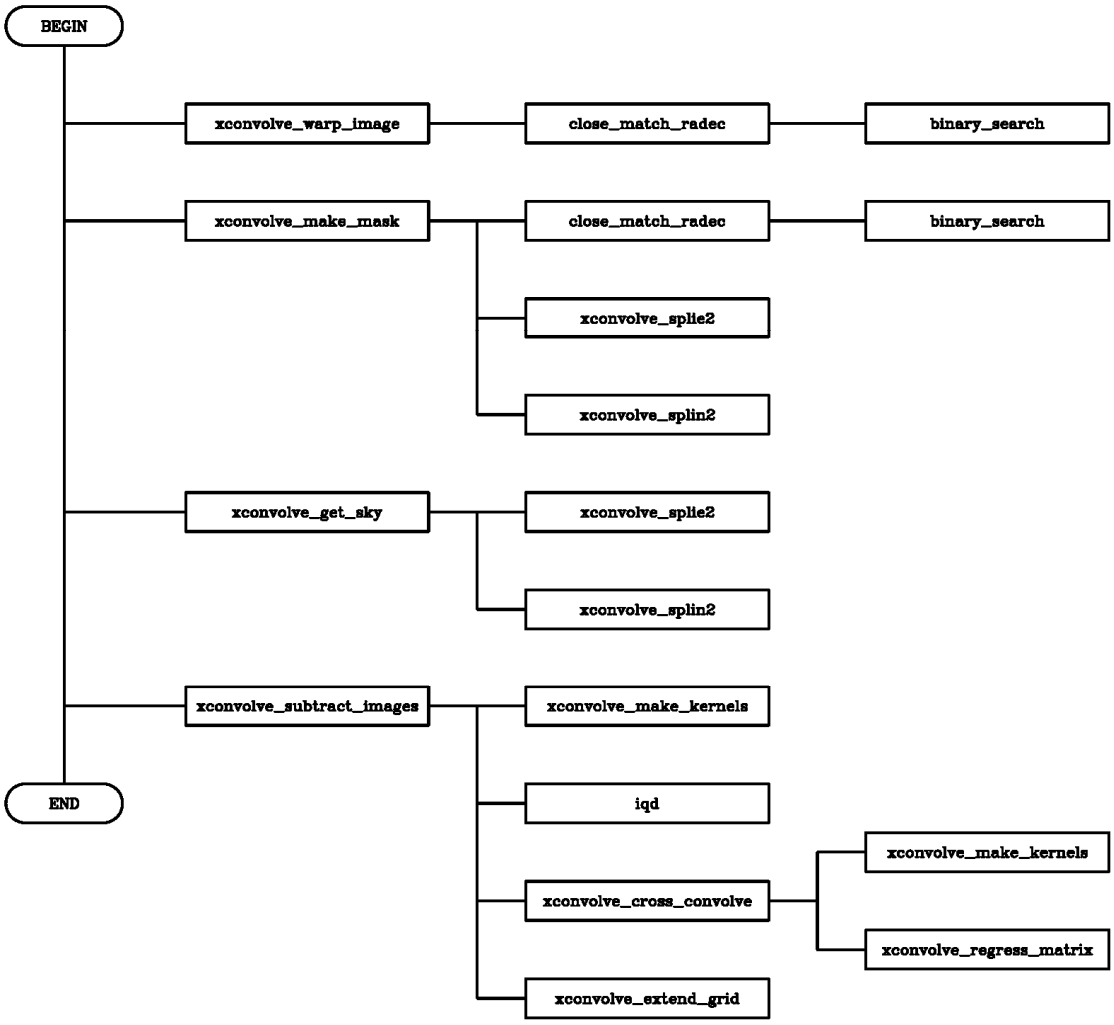

Figure 1. Call tree for XCONVOLVE image subtraction code.