ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XIV
ASP Conference Series, Vol. 347, 2005
P. L. Shopbell, M. C. Britton, and R. Ebert, eds.

Montage: An Astronomical Image Mosaic Service for the NVO

Anastasia Clower Laity, Nate Anagnostou, G. Bruce Berriman, and John C. Good

Infrared Processing and Analysis Center, California Institute of Technology

Joseph C. Jacob and Daniel S. Katz

Jet Propulsion Laboratory, California Institute of Technology

Thomas Prince

California Institute of Technology

Abstract. Montage is a software system for generating astronomical image mosaics according to user-specified size, rotation, WCS-compliant projection and coordinate system, with background modeling and rectification capabilities. Its architecture has been described in the proceedings of ADASS XII and XIII (Berriman et al. 2003, 2004). It has been designed as a toolkit, with independent modules for image reprojection, background rectification and co-addition, and will run on workstations, clusters and grids. The primary limitation of Montage thus far has been in the projection algorithm. It uses a spherical trigonometry approach that is general at the expense of speed. The reprojection algorithm has now been made 30 times faster for commonly used tangent plane to tangent plane reprojections that cover up to several square degrees, through modification of a custom algorithm first derived for the Spitzer Space Telescope. This focus session will describe this algorithm, demonstrate the generation of mosaics in real time, and describe applications of the software. In particular, we will highlight one case study which shows how Montage is supporting the generation of science-grade mosaics of images measured with the Infrared Array Camera aboard the Spitzer Space Telescope.

1. Introduction

Montage builds science-grade image mosaics from any input WCS-compliant data. Separate modules perform reprojection of input images to a common projection, background modeling and rectification, and image co-addition. Montage is portable and highly parallelizable, and can be run in multi-processor or grid environments as well as on a desktop computer. The modular design provides great flexibility to end users. One example is provided by the Spitzer Wide-area Infrared Extragalactic Survey (SWIRE), which uses Montage in its processing pipeline. SWIRE has a wide range of ancillary observations taken using ground- and space-based telescopes, with a variety of image parameters (rotation, projection type, pixel scale, etc.). The SWIRE team uses Montage to re-project and mosaic ancillary data into the same tiling scheme and pixel scale as the mosaics

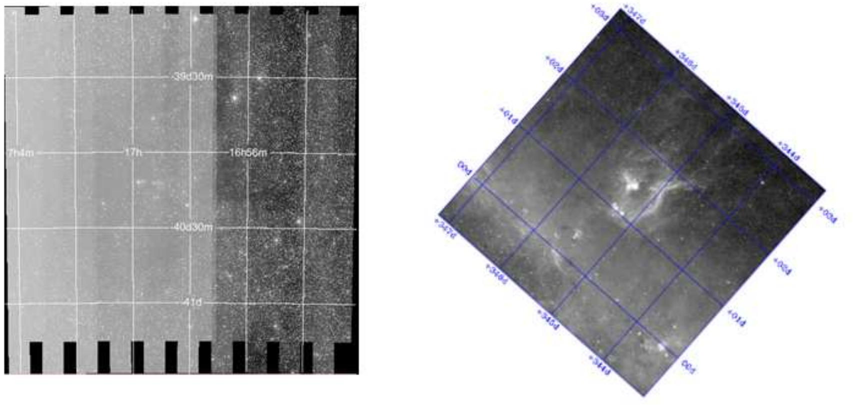

Figure 1. Original 2MASS and MSX images

created by the Spitzer pipeline. This allows optical ancillary data to be easily compared to Spitzer data, facilitating multi-wavelength source extraction and quick recognition of high-redshift objects.

2. Enhancements to Montage Version 2.2: Fast Reprojections and Arbitrary Image Sizes

The latest release of Montage, version 2.2, contains several significant enhancements upon the initial public release, v1.7, and can be downloaded from: http://montage.ipac.caltech.edu

Version 1.7 supported photometrically and astrometrically accurate image reprojections, at the expense of processing speed. The reprojection algorithm, while general, was slow because it projected both input and output pixels onto the celestial sphere in order to calculate the overlap of these pixels on the sky. Version 2.2 of Montage includes a new fast reprojection module for tangent-plane reprojections. mProjectPP has been developed in collaboration with the Spitzer Science Center, and is based on the Mopex algorithm (http://ssc.spitzer.caltech.edu/postbcd/doc/mosaicker.pdf), which uses plane-to-plane solutions to calculate the overlap entirely in pixel space, instead of projecting pixels onto the celestial sphere. This produces a significant speed-up; for 2MASS Atlas images, the improvement is a factor of 20.

Over a small enough area (up to several degrees), many non-tangent plane projections can be approximated by a TAN header with distortion parameters. Montage v2.2 analyzes a FITS header in any projection and determines the tolerances on an equivalent, distorted-TAN projection. It outputs a header template which can be used in conjunction with the fast reprojection module to speed up transformations to or from non-TAN projections.

3. Application of Montage: Creating a 2MASS/MSX Mosaic

This section walks the user through the creation of a 3-color mosaic using data from the 2MASS and MSX missions. On the left of Figure 1 is a mosaic of 170 2MASS Atlas images, with no background rectification, covering a total area of about 3 square degrees. On the right is a MSX A-band (8 μm) image retrieved using IRSA's MSX image server, covering about 2.4 square degrees and in a Cartesian projection. In the following walk-through, each Montage command is preceded with a ">", and is followed by the output of the module. Each Montage module prints a structured output message to stdout, of the form [struct stat="*status*", key1="*val1*", ... , keyn="*valn*"].

Step One: Setup metadata tables and header templates. First, create tables of image metadata (WCS information, FITS geometry, etc.) for the Montage modules to read, instead of having to repeatedly open and close FITS files:

```
>mImgtbl raw_K raw_K.tbl
[struct stat="OK", count=170, failed=0, nooverlap=0]
```

Call mMakeHdr to create a header template that completely encloses all the 2MASS images:

```
>mMakeHdr raw_K.tbl template.hdr
[struct stat="OK", count=170, clon=254.587292, clat=-40.25175
3, lonsize=2.353611, latsize=2.450000, posang=359.891421, lon
1=256.154189, lat1=-41.468162, lon2=253.014309, lat2=-41.4636
21, lon3=253.076184, lat3=-39.014964, lon4=256.104469, lat4=
-39.019343]
```

The 2MASS images are in a tangent-plane projection, and so can use the fast reprojection module. To speed up the MSX reprojection, however, we need to create a distorted-TAN header for the MSX data:

```
>mGetHdr raw_MSX/msx_4deg.fits msx.hdr
[struct stat="OK", ncard=23]
>mTANHdr -c eq msx.hdr msxtan.hdr
[struct stat="OK", fwdxerr=0.00351429, fwdyerr=0.0054
6297, fwditer=51, revxerr=0.00335636, revyerr=0.03825
81, reviter=9]
```

Step Two: Image reprojection. Launch the 2MASS image reprojections by calling the reprojection executable, mProjExec, and using the "-f" flag to instruct it to use fast reprojection:

```
>mProjExec -f -p raw_K raw_K.tbl template.hdr proj_K
stats_K.tbl
[struct stat="OK", count=170, failed=0, nooverlap=0]
```

Next, call the fast reprojection module directly on the MSX data, instructing it to use the distorted-TAN template header instead of reading the native MSX FITS header:

```
>mProjectPP -i msxtan.hdr raw_MSX/msx_4deg.fits
final_MSX.fits template.hdr
[struct stat="OK", time=6082]
```

Once the reprojections are complete, it is necessary to regenerate the image metadata tables, as the FITS geometry has changed:

```
>mImgtbl proj_K proj_K.tbl
[struct stat="OK", count=170, badfits=0]
```

Step Three: Background rectification. The background rectification process matches each image's background to its surrounding images, globally minimizing the inter-image differences. First, create a table which lists the files that overlap each other and gives each overlap pair a unique identifier:

```
>mOverlaps proj_K.tbl diff_K.tbl
[struct stat="OK", count=454]
```

Second, create "difference" images for each overlap region by subtracting FITS files from each other:

```
>mDiffExec -p proj_K diff_K.tbl template.hdr diff_K
[struct stat="OK", count=454, failed=0]
```

Third, call mFitplane on each difference image to find the plane that best fits each one:

```
>mFitExec diff_K.tbl fits_K.tbl diff_K
[struct stat="OK", count=454, failed=0, warning=0,
missing=0]
```

Fourth, using the information found by mFitplane, calculate what planes need to be removed from each image in order to globally minimize the background differences:

```
>mBgModel proj_K.tbl fits_K.tbl corrections_K.tbl
[struct stat="OK"]
```

Fifth and finally, call mBackground on each projected image to subtract the plane calculated by mBgModel:

```
>mBgExec -p proj_K proj_K.tbl corrections_K.tbl corr_K
[struct stat="OK", count=170, nocorrection=0, failed=0]
```

Step Four: Coaddition.
The last step for the two bands of 2MASS data is to coadd them into mosaics:

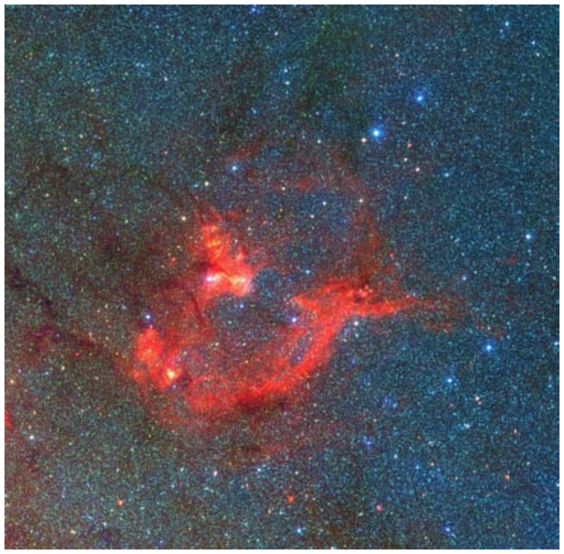

Figure 2. Final 3-color mosaic: MSX A-band is red, 2MASS K-band is green, and 2MASS J-band is blue.

```
>mAdd -e -p corr_K proj_K.tbl template.hdr final_K.fits
[struct stat="OK", time=144]
```

Step Five: Creating a 3-color JPEG image.
After cropping the edges out of each mosaic, call mJPEG (another new module; not included in the public release, but available on request) to create a 3-color JPEG image from the 3 FITS files. The user assigns a color to each image and inputs the desired color-stretch, which can be found using a visualization tool such as IRSA's OASIS. The output of mJPEG is shown in Figure 2.

```
>mJPEG -red final_MSX_crop_4.fits 0% 99.95% 2
-green final_K_crop_4.fits 0% 99.3% 2
-blue final_J_crop_4.fits 0% 99.4% 2
-out jpeg/r99.95_g99.3_b99.4_crop_4.jpg
[struct stat=OK]
```

Acknowledgments. Montage is funded by NASA's Earth Science Technology Office, Computational Technnologies Project, under Cooperative Agreement Number NCC5-626 between NASA and the California Institute of Technology.

References

Berriman, G. B., et al. 2003, in ASP Conf. Ser., Vol. 295, ADASS XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 343

Berriman, G. B., et al. 2004, in ASP Conf. Ser., Vol. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 593