# Multi-Wavelength Image Space: Another Grid-Enabled Science

Roy **Williams** [1]
Bruce **Berriman**[2]
Ewa **Deelman**[3]
John **Good**[2]
Joseph **Jacob**[4]
Carl **Kesselman**[3]
Carol **Lonsdale**[2]
Seb **Oliver**[5]
Thomas A. **Prince**[1]

[1]CACR, California Institute of Technology, USA
[2]IPAC, California Institute of Technology, USA
[3]ISI, University of Southern California, USA
[4]JPL, California Institute of Technology, USA
[5]Astronomy Centre, University of Sussex, UK

**Abstract**

We describe how the Grid enables new research possibilities in astronomy through multi-wavelength images. To see sky images in the same pixel space, they must be projected to that space, a compute-intensive process. There is thus a virtual data space induced that is defined by an image and the applied projection. This virtual data can be created and replicated with Planners and Replica catalog technology developed under the GriPhyN project. We plan to deploy our system (MONTAGE) on the US Teragrid. Grid computing is also needed for ingesting data -- computing background correction on each image -- which forms a separate virtual data space. Multi-wavelength images can be used for pushing source detection and statistics by an order of magnitude from current techniques; for optimization of multi-wavelength image registration for detection and characterization of extended sources; and for detection of new classes of essentially multi-wavelength astronomical phenomena. The paper discusses both the grid architecture and the scientific goals.

## 1. Introduction: Multi-Wavelength Astronomy

In this paper we discuss a new and largely unexplored paradigm in astronomy: the multi-wavelength image. We will describe both scientific goals, and also how grid computing can open this new field. The term multi-wavelength refers to both a scientific and a sociological fusion: we are federating surveys from different wavelengths (infrared, optical, radio, etc), but also from different groups of astronomers, who have traditionally specialized by wavelength, based on the instrument with which they observe, rather than by the physical processes actually occurring in the Universe.

When a sky survey is made, the telescope images different parts of the sky at different times, creating an overlapping mosaic of images. These images are calibrated and placed on a common intensity scale, corrected for pixel sensitivity variations, so that the released product is corrected for instrumental effects. Traditionally, the images are converted to star/galaxy catalogs, and much of the scientific knowledge is mined from these catalogs. The images are typically analyzed on a one-by-one basis, if at all. However, in recent years we have begun to see an effort to federate these catalogs across the wavelengths, breaking down the traditional specialties of astronomical communities, which are based on the instrument with which they observe. There are both knowledge-based resources, and big-data projects[1] that have made good progress with federating catalogs across the wavelengths.

In this paper, we consider a new paradigm for mining knowledge from the images of the sky surveys: by federating the images directly to a uniform pixel space[2][3], then doing pattern matching in the multi-wavelength space. In order to do this, we must 'resample' images to the common pixel space, a process that results in a loss of data quality, but which we hope will benefit greatly from federation.

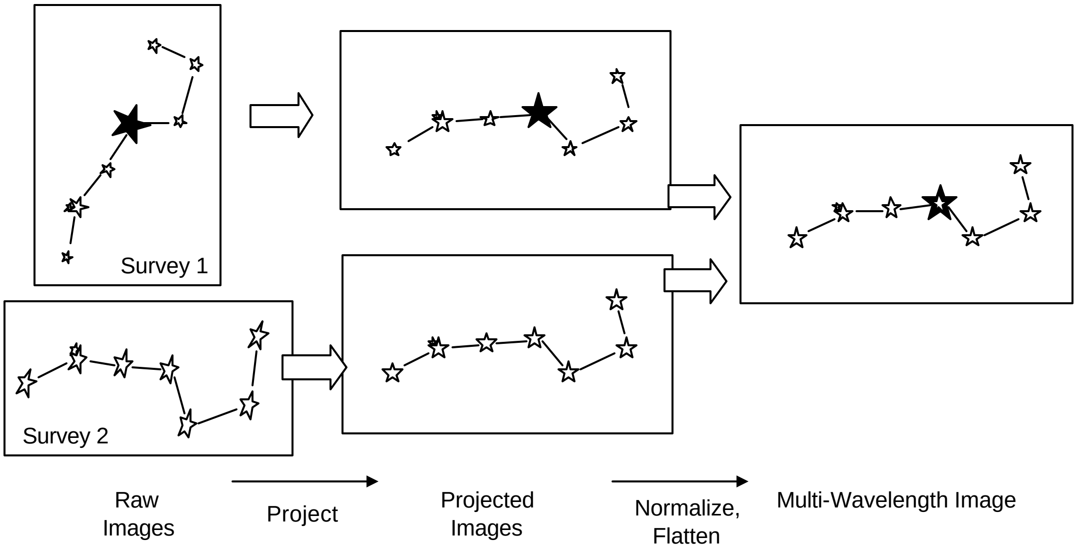

Figure 1. Multi-wavelength image composition. (Left) Images from two surveys at different wavelengths are stretched and distorted with respect to each other, and also show different physics (eg. black star in survey 1). (Middle) They are pro jected to the same pixel space, and (Right) They are normalized and flattened into a single, co-registered multi-wavelength image.

The MONTAGE project has been funded by NASA, and the request management component through the NSF National Virtual Observatory, to build Grid-enabled image mosaicking services. This paper explains our initial view of the reasoning and architecture. The services will offer simultaneous, parallel processing of multiple images to enable fast, deep, robust source detection in multi-wavelength image space. These services have been identified as cornerstones of the National Virtual Observatory. We intend to work with both massive and diverse image archives: the 10 Tbyte 2MASS (infrared [4]), the 3 Tbyte DPOSS (optical[5]), and the much larger SDSS[6] optical survey as it becomes available. There are many other surveys of interest.

A browseable, education-oriented prototype of the multiwavelength sky is already available at VirtualSky.org [7].

## 2. Scientific Goals

Some of the goals of multiwavelength image federation are as follows:

- **Fainter sources**: We will extend source detection methods up to detect objects an order of magnitude fainter than currently possible. A group of faint pixels may register in a single wavelength at the two-sigma level (meaning there may be something there, but it may also be noise). However, if the same pixels are at two-sigma in two other surveys, then the overall significance may be boosted, indicating an almost certainty of the existence of signal rather than just noise. We can go fainter in image space because we have more photons from the combined images and because the multiple detections can be used to enhance the reliability of sources at a given threshold.
- **Spectrophotometry :** characterizing the spectral energy distribution of the source through "bandmerge" detections from the different wavelengths.
- **Position Optimization** between sources at different wavelengths. A prerequisite for finding differences between sky images is the perfect mutual positioning of sources; without it the difference field is confused by every star appearing as a positive-negative pair. We will focus on reconciling images with very different inherent spatial resolution.
- **Extended Sources**: Robust detection and flux measurement of complex, extended sources over a range of size scales. Larger objects in the sky (eg. M31, M51) may have both extended structure (requiring image mosaicking) and a much smaller active center, or diffuse structure entirely. Finding the relationship

between these attributes remains a scientific challenge. We will be able to combine multiple instrument imagery to build a multi-scale, multi-wavelength picture of such extended objects. It is also interesting to make statistical studies of less spectacular, but extended, complex sources that vary in shape with wavelength.

- **Image Differencing**: Differences between images taken with different filters can be used to detect certain types of sources. For example, planetary nebulae (PNe) emit strongly in the narrow H$\alpha$ band. By subtracting out a much wider band that includes this wavelength, the broad emitters are less visible and the PNe is highlighted.
- **Time Federation**: A trend in astronomy is the *synoptic* survey, where the sky is imaged repeatedly to look for time-varying objects. MONTAGE will be well-placed for mining the massive data from such surveys.
- **Essentially Multi-wavelength Objects**. We will use the new multi-wavelength images to specifically look for objects that are not obvious in one wavelength alone. Quasars were discovered in this way by federating optical and radio data. We hope for discovery of new classes of 'essentially multi-wavelength' objects. We will make sophisticated, self-training, pattern recognition sweeps through the entire image data set. An example is a distant quasar so well aligned with a foreground galaxy to be perfectly gravitationally lensed, but where the galaxy and the lens are only detectable in images at different wavelengths.

# 3. Grid Architecture

## 3.1. Data Pipeline

The architecture is based on the Grid paradigm, where data is fetched from the most convenient place, and computing is done at any available platform, with single sign-on authentication to make the process practical. We will also rely on the concept of 'virtual data', the idea that data requests can be satisfied transparently whether the data is available on some storage system or whether is needs to be derived in some way. With these architectural drivers, we will be able to provide customized, high-quality data, with great efficiency, to a wide spectrum of usage patterns.

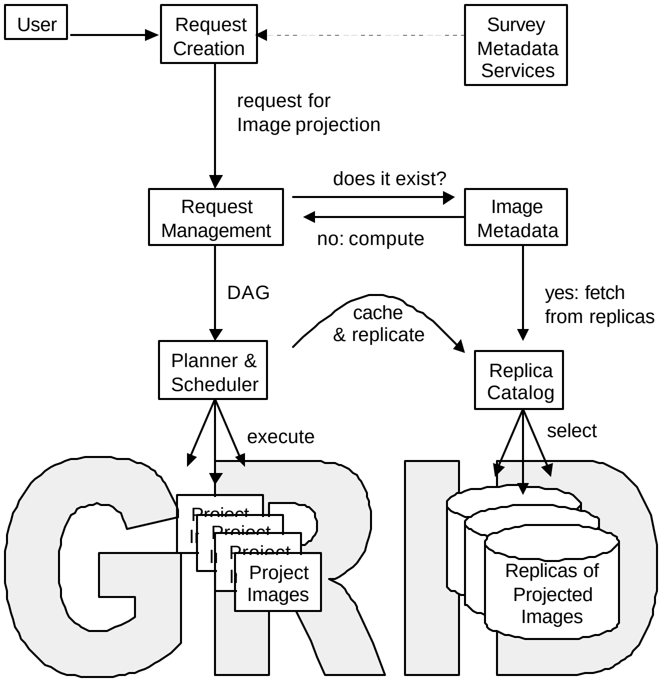

Figure 2: MONTAGE Architecture. After a user request has been created and sent to the Request Manager, part of the request may be satisfied from existing (cached) data. The Image Metadata (IM) system looks for a suitable file, and if found, gets it from the distributed Replica Catalog (RC). If not found, a suitable computational graph (DAG) is assembled and sent to be executed on Grid resources. Resulting products may be registered with the IM and stored in RC. The user is notified that the requested data is available until a specified expiry time.

At one end of the usage spectrum is the scientist developing a detailed, quantitative data pipeline to squeeze all possible statistical significance from the federation of multiple image archives, while maintaining parentage, rights, calibration, and error information.

In this case the request document would have no defaults. The specification includes the background estimation -- with its own fitting function and masking, as well as cross-image correlation; custom projection from sky to pixel grid, the details of the resampling and flux preservation is custom, and specification of how pixels are to be coadded, and how blanked/masked pixels should be treated. Currently this request specification is made with a hierarchical keyword-value paradigm, for example "projection=TAN" to specify tangent-plane reprojection.

In this case, the scientist would have enough authorization that powerful computational resources can be brought to bear, each processor finding the nearest replica of its input data requirements, and the output being hierarchically collected to a final composite. Such a product will require large computational, data, and bandwidth resources from the Teragrid[8], and the result will be published in a peer-reviewed journal as a scientifically authenticated, multi-wavelength representation of the sky.

Other users will have less stringent requirements for the way in which image mosaics are generated. They will build on a derived data product such as described above, perhaps using the same background model, but with the resampling different, or perhaps just using the derived product directly. When providing users with the desired data, we want to be able to take advantage of the existing data products and produce only the necessary missing pieces. It is also possible, that it may take longer to access the existing data rather than performing the processing. These situations need to be analyzed in our system and appropriate decisions need to be made.

## 3.2. Replica Management

Management of replicas in a data pipeline means that intermediate products are cached for reuse: for example in a pipeline of filters ABC, if the nature of the C filter is changed, then we need not recompute AB, but can use a cached result. Replica management can be smarter than a simple file cache: if we already have a mosaic of a certain part of the sky, then we can generate all subsets easily by selection. Simple transformations (like selection) can extend the power and reach of the replica software. If the desired result comes from a series of transformations, it may be possible to change the order of the transformations, and thereby make better use of existing replicas.

Currently, data requests are built with a collection of keyword-value pairs. The initial invocation of the replica manager will not use a replica unless either (a) the request we are trying to satisfy is identical to a previous request, or (b) the only difference is that the original request has computed a superset of the pixels wanted by the new request. The replica manager will have a purge policy in the future, but currently is cleared by a human.

## 3.3. Virtual Data

Further gains in efficiency are possible by leveraging the concept of 'virtual data' from the GriPhyN project[9]. The user specifies the desired data using domain specific attributes and not by specifying how to derive the data, and the system can determine how to efficiently build the desired result. Replica management is one strategy, choosing the appropriate computational platforms and input data locations is another.

The interface between an application programmer and the Virtual Data management systems (VDMS) now being deployed, is the definition of what is to be computed, which is called a Virtual Data Request (VDR). The VDR specifies what data is to be fetched -- but not the physical location of the data, since there may be several copies and the VDMS should have the liberty to choose which copy to use. Similarly the VDR specifies the set of objects to compute, but not the locations of where the computations will take place.

## 3.4. Request Management

The MONTAGE system consists of a set of services that allow users to build large or small image projection products. Requests for on-demand processing can be submitted through an astronomical information system such as yourSky [10], Oasis [11], or VirtualSky[7], or larger batch orders submitted by editing an XML file containing the request. The Request Management System is responsible for keeping the state of a request and reporting diagnostics as it progresses from submission, to stages of execution, to output in temporary storage, to expiration. Job pausing and cancellation will also be possible.

The interpreter takes the XML request and translates it into an abstract directed acyclic task graph (aDAG). The aDAG specifies the computations which need to take place (transformations) and the data needed for the computation without specifying the physical location of the data. The abstract DAG represents the sequence of operations needed to project a specific image with a specific projection.

## 3.5. Replica Catalog

The Globus[12] data grid toolkit provides a layered replica management architecture. At the lowest level is a *Replica Catalog* that allows users and applications to register files as logical collections and provides mappings between logical names for files and collections and the storage system locations of file replicas. A low-level API is provided as an interface to the catalog. This API can be used by higher-level tools such as the request manager that selects among replicas based on network or storage system performance.

The Replica Catalog provides simple mappings between logical names for files or collections and one or more copies of those objects on physical storage systems. The catalog registers three types of *entries*: logical collections, locations, and logical files. A *logical collection* is a user-defined group of files. Users might often find it convenient and intuitive to register and manipulate groups of files as a collection, rather than requiring that every file be registered and manipulated individually. *Location* entries in the replica catalog contain the information required for mapping a logical collection to a particular physical instance of that collection. Each location entry represents a complete or partial copy of a logical collection on a storage system. The replica catalog also includes optional entries that describe individual *logical files*. Logical files are entities with globally unique names that may have one or more physical instances.

## 3.6. Planning and Scheduling

This abstract representation is sent to the planner, which combines information gathered about the data location(s), location(s) where the transformations can take place and constructs a concrete DAG. Currently, planners are being developed as part of the GriPhyN project. In our view, the user specifies the desired data product by specifying the application-specific attributes such as projections (see Section 4). An application-specific component interprets the user's request and determines the necessary operations needed to produce the data product (in a form of a DAG). The planner takes the DAG, consults the replica catalog to determine the location of the input data and the transformation catalog[13] to locate where the transformations can take place (which particular hosts). The planner also needs to query the Grid information services, for example the Globus MDS[12] to determine the availability and load of the resources. Information about the network performance can be obtained from NWS[14].

Given the performance information as well as data and computation location, the planner needs to make the "best choice". At this time GriPhyN is evaluating the standard AI planning techniques for suitability. Performance is only one factor in making the "best" plan, reliability needs to be taken into account as well. Even if the estimated performance of a given plan is optimal, one might still want to pick a plan that is not expected to perform as well but may have a higher chance of success. The current invocation of the planner is given the location of the dataset, and simply tries to find the nearest computational resource to that data, based on a priority list.

Once the plan is made, the DAG can be sent to the executor which will schedule the specified data movements and computations. In this project, as in GriPhyN, we will use the Condor-G[15] server and the associated DAGMan[15] which analyses the DAG and then submits jobs in the manner in which they are defined. These jobs may run locally or on a Condor pool or via a GRAM interface to a resource, where Globus is available. If any of the tasks in the plan fail then the execution of the DAG execution is halted and the error is reported to the user. Condor-G has a restart facility available by which a job can be resubmitted and it can continue from where it stopped. If all the jobs in a plan succeed then the data is made available to the user. In the end the application metadata catalog as well as the replica catalog might be updated to reflect the availability of the new data.

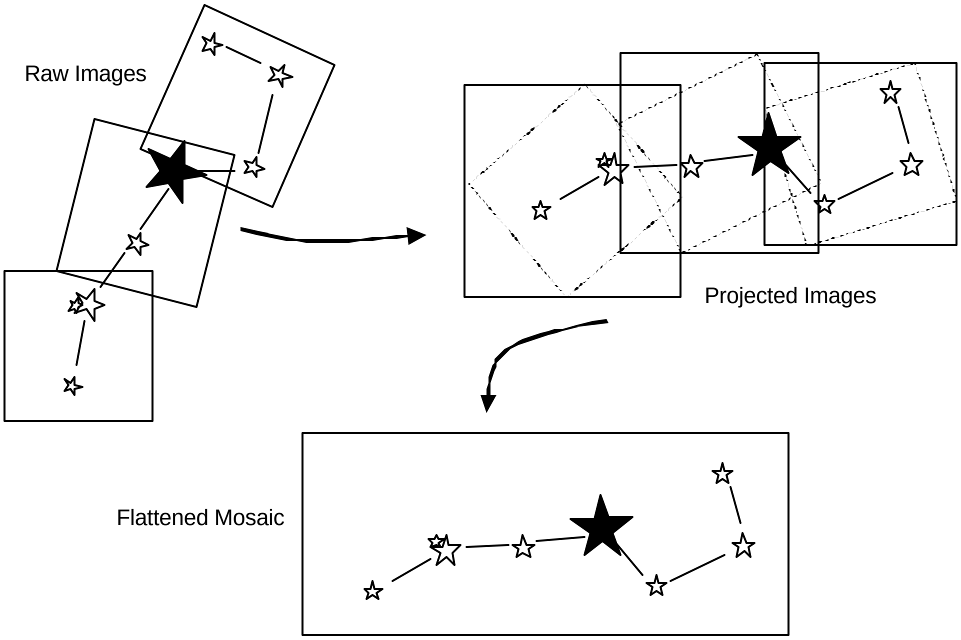

Figure 3: Projection and flattening. The raw images may have been taken on different dates. Each is projected and normalized separately, and some applications may use these projected images directly. The flattening process makes a single wider area mosaic from the projected images, using a (customizable) averaging process in the overlap area.

# 4. Astronomical Image Projection

## 4.1. Projections

In the case of astronomical image mosaicking, the VDR will specify the survey from which data is to come, the component color filters (G, R, I etc) of that survey and the region of the sky that is to be delivered. Also necessary is the projection of sky to pixels that will be used, selected from the standard World Coordinate System (WCS) set [16], the method of resampling to be used, the way in which image overlap will be handled (first pixel, average, weighted average, etc.), and the type of normalization to be used (polynomial order and masking strategy).

Each image from the survey will be resampled into a separate image in the output plane, which we shall call a projected image. The projection is defined by:

- The projection type (enumeration of 30 types in the WCS system)
- Projection parameters (0-3 floats)
- The projection center and rotation on the sky (3 floats)
- The projection scale in each direction (2 floats)
- The number of pixels in each direction (2 integers)

The process is shown in Fig. 1. Given a region of the sky, images are chosen from each survey, and they are projected to common pixel grid and combined.

### 4.2. Parallelizing Image Projection

Parallelizing the creation of astronomical image mosaics can be done in two ways that may be combined; we can parallelize on the input data or on the output data. Parallelizing on input (PI) means that each image of the survey goes to a different processor, and creates its own layer of the final mosaic. The flattening process would then require an all-to-all data movement that could be very inefficient if the output mosaic is very large. Parallelizing on output (PO) means that each processor is responsible for a subset of the final output plane, and opens the files it needs from the input survey. In this case, there may be inefficiency on the input side, as some files could be opened and ingested many times.

The parallelization methods should be combined for maximum efficiency. The PI process is used on tightly connected clusters to minimize overhead on input. The PO process is used to distribute large mosaics across a large number of tight clusters.

### 4.3. Projected Images

Flattening of projected images is illustrated in Figure 4. Depending on the final result, projected images may or may not be ‘flattened’ like this into a single mosaic. This is because data is lost in the flattening process, and this may not be desirable for the particular astronomical application. For example, each image of the original survey is presumably taken at a different epoch (of time), and flattening will put pixels from one epoch with those from another. The scientist may or may not want to do this: those studying distant galaxies may not care, but those searching for minor planets surely will.

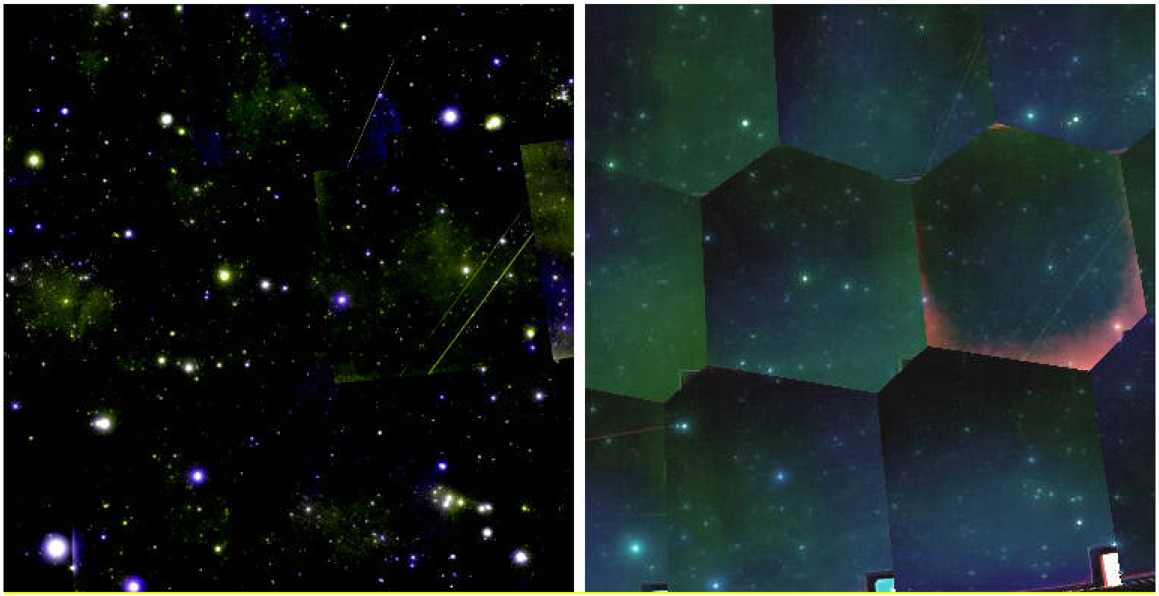

Figure 4: The same part of the sky with and without normalization (background subtraction). This is about 120 square degrees from the DPOSS optical survey, as delivered by VirtualSky.org [11].

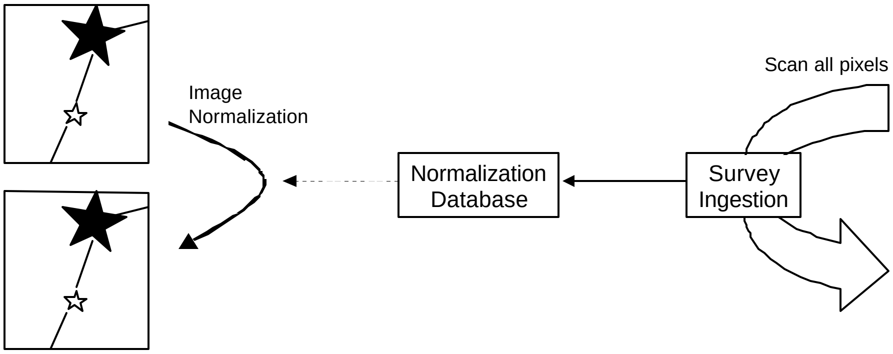

Figure 5: Image normalization involves computing a model of statistical background, to separate from signal. Best results are achieved with a global optimization of the background model, meaning that all images must be processed to compute a good normalization (Right). The resulting polynomial (a few hundred bytes per image) are stored in a database (Center). Images to be normalized read the database for the appropriate polynomial.

## 4.4. Image Normalization

Figure 4 shows the necessity of the normalization process. A statistical model is made of the pixel brightness, so that astrophysical signals and noise can be separated. Noise may come from known instrumental effects, or from unknown atmospheric or other effects. The best model for this noise may be achievable on an image by image basis, but the more challenging case for data-computing is where a global optimization is needed. If the noise comes from the Earth's atmosphere in a drift scan observation, then the best noise model comes from comparing sets of adjacent images in a global fashion. The algorithm is to first mask out bright pixels from stars (because we are modeling noise only), then fit a polynomial across the image to the local median of the darker pixels. At this point, a global optimization may be made across all images. There are different ways to build the normalization models; we expect a power user to want a custom normalization, but a majority to use an existing model.

In MONTAGE, we expect to distribute both the computation of the normalization model and its use (by the mosaicking service), as illustrated in Figure 5. The models are stored in a central database, and write permission to the database is both carefully controlled, but also available to the distributed machines that are ingesting the survey. Read access will be less carefully controlled.

# 5. Required Algorithms

As noted above, the flattening of projected images to a mosaic will depend on context. It includes (1) literal coadding of pixel data from each input image into a final, re-projected, coadded image; and (2) virtual stacking of images such that the individual image pixel data, and any re-projected images are simultaneously maintained along with the virtual coadd.

The requisite techniques to achieve the goals outlined above include:

1. Optimal geometrical alignment and stacking of multiple images taken at different wavelengths from different observing facilities and with different point spread functions (PSFs), including corrections for geometrical distortions, using (a) header WCS information, (b) correlations between detected point sources, and (c) more general methods that do not rely on extracted lists of point sources.

2. Stacking of multi-wavelength images to optimize the reliable detection and photometry of the sources of interest. Investigation of the optimum weighting by the spectral energy distributions (SEDs) of the sources of interest and/or the sources of noise present in the images. This could be a generalization of the methods described by Szalay et al.[2] and Hopkins et al.[3], and such a super-image will have a signal that represents the minimum variance estimate of the flux of a source of a specific SED, and can be tuned for various SEDs.

3. Development of a robust and optimized approach to simultaneously estimate the flux and positions of sources given the multi-wavelength data sets, ie. simultaneous solution of steps (1) and (2) above.

4. Development of methods to query the image stacks which have been tuned for given experimental conditions and science goals. To include: searches for outliers in shape and/or color that are artifacts or rare sources; searches for sources that have inherently different structures at different wavelengths; study of sources on multiple size scales significantly larger than typical individual images; optimization for diffuse and low surface brightness emission structures; pattern recognition routines optimized for wavelength-dependent structures.

# 6. Acknowledgements

We would like to acknowledge support from NASA under NASA/Goddard Cooperative Agreement NCC5-626, and from NSF under University Agreement No. 8111-53652 and Cooperative Agreement AST-0122449.

# 7. References

[1] Rutledge, R E.; Brunner, R J.; Prince, T A.; Lonsdale, C, *XID: Cross-Association of ROSAT/Bright Source Catalog X-Ray Sources with USNO A-2 Optical Point Sources* , Ap. J Supp., **131**, 335

[2] A. Szalay., A. J. Connolly, G. P. Szokoly, *Simultaneous Multicolor Detection of Faint Galaxies in the Hubble Deep Field*, Astron. J. **117** 68, (astro-ph/9811086)

[3] A. M. Hopkins, C. J. Mille, A. J. Connolly, C. Genovese,R. C. Nichol, L. Wasserman, *A new source detection algorithm using FDR2002,* accepted for publication by Astron. J.(astro-ph/0110570)

[4] 2MASS: The Two-Micron All-Sky Survey, http://www.ipac.caltech.edu/2mass/

[5] DPOSS: The Digitized Palomar Observatory Sky Survey, http://www.astro.caltech.edu/~george/dposs/

[6] SDSS: The Sloan Digital Sky Survey, http://www.sdss.org/

[7] R. D. Williams, *Virtual Sky: Multiwavelength sky browsing,* http://www.virtualsky.org/ and http://www.npaci.edu/envision/v17.4/education.html

[8] Teragrid: http://www.teragrid.org

[9] GriPhyN: *Grid Physics Network,* http://www.griphyn.org/

[10] J. Jacob, *yourSky: An Interface to the National Virtual Observatory Mosaicking Code*, NASA Science Information Systems Newsletter, Issue 60, ed. B.J. Sword, July, 2001, http://www-sisn.jpl.nasa.gov/issue60/article_yourSky.html.

[11] *Oasis; On-line Archive Science Information Services;* http://irsa.ipac.caltech.edu/applications/Oasis/

[12] I. Foster, C. Kesselman et. al., *Globus: fundamental technologies needed to build computational grids*, http://www.globus.org/

[13] E. Deelman, C. Kesselman, S. Koranda, A. Lazzarini, and R. Williams, *Applications of Virtual Data in the LIGO Experiment*, Proc. Fourth Int. Conf. on Parallel Processing and Applied Math. (PPAM'2001). Lect. Notes Comp. Sci. (to appear), http://www.isi.edu/~deelman/ppam01.pdf

[14] *NWS: Network Weather Service*: http://nws.npaci.edu/NWS/.

[15] *Condor: High Throughput Computing*, http://www.cs.wisc.edu/condor/

[16] E. W. Greisen and M. Calabretta, *Representations of Celestial Coordinates in FITS*, http://www.atnf.csiro.au/people/mcalabre/WCS.htm.