Fusion, Visualization and Analysis Framework for Large, Distributed Data Sets[1]

Joseph C. Jacob
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Mail Stop 126-234
Pasadena, CA 91109-8099
818-354-0673
Joseph.Jacob@jpl.nasa.gov

Lucian Plesea
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Mail Stop 126-234
Pasadena, CA 91109-8099
818-354-3928
Lucian.Plesea@jpl.nasa.gov

Abstract—In this paper a framework is presented for extracting information content from modern sky surveys, which have archived multiple terabytes of data in various wavelengths and at various resolutions. The proposed framework includes new technology that addresses the massive size and geographically distributed nature of these data sets. Also included is automated support for combining data sets from multiple archives and for relating sky catalogs to the image data. In addition, tools are provided for efficiently exploring images that are hundreds of gigabytes or even multiple terabytes in size. The proposed framework and "data agile" applications described here are essential in the modern era of astronomy because images of this size far exceed the current capabilities of conventional image analysis tools used in the astronomical community.

TABLE OF CONTENTS

1. INTRODUCTION

One objective of the National Virtual Observatory [1] initiative is to federate various sky surveys hosted at institutions geographically distributed across the United States. Ultimately this concept will be expanded into a Global Virtual Observatory including sites in other countries. The larger sky surveys such as the Two Micron All Sky Survey (2MASS) [2] in the infrared and the Digital Palomar Observatory Sky Survey (DPOSS) [3] in the visible each contain image and catalog data that exceeds the terabyte level in size. The large size and distributed nature of these data sets poses a significant information technology challenge that exceeds the current capabilities of visualization software, compute power, and network infrastructure. The fundamental problem that is addressed in this paper is how to explore effectively and extract information from these enormous data sets.

This paper describes the system that we are developing for high performance exploration of large data sets. The software can access the image data either locally or from remote servers, combine them on the fly, and project the resulting image on single screen or PowerWall [4] displays, described in Section 5. The data fusion can be done in a number of ways to produce images with richer information content than is contained in the individual components. Some examples are as follows:

- High-resolution insets of celestial objects or regions of special interest overlaid on top of lower resolution images of larger regions of the sky.
- Images from surveys in different wavelengths combined to produce novel multi-spectral views of the sky.
- Vector overlays or synthetic maps derived from catalog data combined with real images of the sky.

Latitude and longitude information provided along with the image data is used to position correctly and to scale the data sets relative to each other. Any number of data sets may be combined in this way, with a slight degradation in performance with each image added.

This software has been used to overlay high-resolution insets of 2MASS and DPOSS data at 1 arc second on top of a mosaic constructed from images captured by the Infrared Astronomical Satellite (IRAS) at 1 arc minute. We are constructing new larger sky mosaics that will require the technology described in this paper for viewing. Although this technology is intended for visualization of sky data, much of it is applicable to other types of data sets such as planetary images (e.g. Synthetic Aperture Radar, Landsat, etc.).

There are a number of excellent utilities that can be used to remotely access sky images at various wavelengths in a web browser. One well-designed system is SkyView [5], which

---

[1] 0-7803-6599-2/01/$10.00 © 2001 IEEE

can generate custom images that meet user specification of position, scale, orientation and survey at wavelengths from radio to gamma ray. SkyView is intended for custom access to relatively small images, typically smaller than the desktop display. No interactive capabilities for exploring the generated images are provided or required due to the small size of the images that are generated. The time required to generate much larger images is prohibitive. We have developed a system that provides similar custom access to images but with no limitation on the size of the image that is generated. Images of size greater than 100 GB or even a terabyte are possible. Furthermore, we provide tools for high performance exploration of these huge images, i.e. smooth pan and zoom from the synoptic view to any part of the data at any resolution. This is a marked improvement over current systems that require issuing separate requests for each static snapshot image that is needed and waiting for each result. Additional improvements are support for high resolution insets and viewing on both single screen and synchronized multi-screen PowerWall displays.

An overview of the proposed framework is provided in Section 2. Our mosaic building application is described in Section 3. The image data formats we use are described in Section 4. Our high-resolution display capabilities are discussed in Section 5. Information about how the system does automatic run time data fusion is provided in Section 6. In Section 7 we discuss how the system can operate in a client-server mode with images served from remote hosts. A description of our capabilities for exploring large images and catalogs is provided in Section 8. A summary is provided in Section 9.

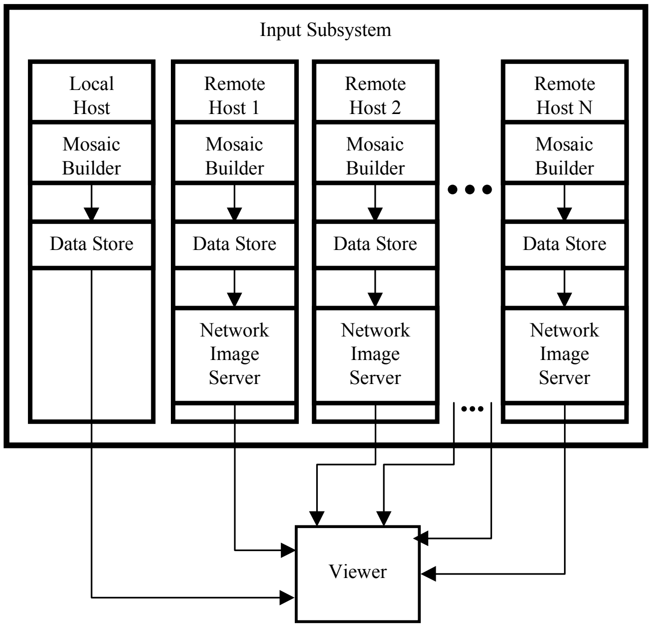


Figure 1. The input subsystem delivers image data to the viewer.

2. SYSTEM COMPONENTS

The system we are developing consists of an input subsystem that delivers image data to a viewer as illustrated

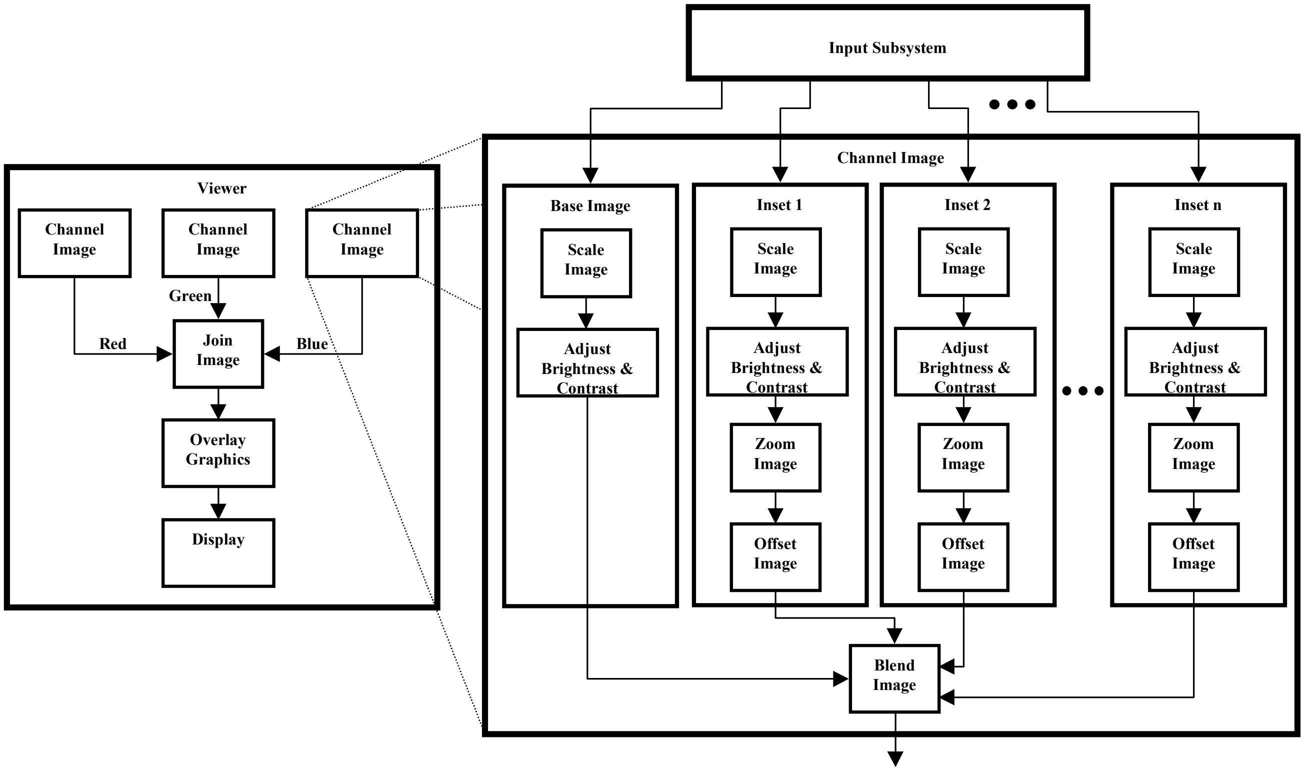


Figure 2. The viewer can automatically map to the red, green and blue video channels an image that is a composite of a base image and a number of inset images.

in Figure 1. The input subsystem has two main software components, the mosaic builder and the network image server. As shown in the figure, the image data may be stored on the local host where the viewer is running or on any number of remote hosts. If the viewer is running on the machine where the image data resides, it reads the image data directly from the data store on that machine. However, if the image data is stored on a remote host, the network image server must be used to read from the data store and serve image data over the network to the viewer.

Figure 2 shows a more detailed flowchart of the viewer used in the system. As shown in the figure, the input subsystem delivers data from any number of input images to the viewer. Based on user specification, the multiple images are combined in different ways to produce a composite image that is sent to the display. A separate channel image may be created for each of the red, green and blue video channels. The channel images are then joined to form the RGB color image that is seen on the display. Each channel image is itself a composite of a base image and multiple inset images. The base image and each of the insets are scaled to fit the data type of the output mosaic. A brightness and contrast adjustment may also be applied to each of the images. The inset images are each zoomed to match the resolution of the base image and are offset to the correct position relative to the base image. This application of zoom factors and offsets is done automatically using position and resolution information that must be provided with each input image. After completion of the join operation to produce the RGB output image, vector graphics may be drawn as an image overlay.

3. LARGE SKY MOSAIC CONSTRUCTION

The virtual observatory software that we are developing can be used both to generate large image mosaics from sky image patches and to view these large images. The mosaicking software is fully automated and can be run in parallel on multiprocessor systems. The input image patches may be in the common Flexible Image Transport System (FITS) format [6], a universally accepted standard for encoding astronomical data. FITS images consist of records containing either pixel values or keyword-value pairs that provide supplementary information about the image. The image patches may be at any resolution, in any coordinate system and projection, and having any data type supported by FITS. The software translates these patches into the user-selected resolution and coordinate system of the output mosaic. The World Coordinate System library [7] is used to convert the input FITS images into a common output coordinate system and projection. The galactic, ecliptic, J2000, and B1950 coordinate systems are supported. The data type of the output pixels in the mosaic may be N-byte integer (any $N > 0$) or floating point. This permits construction of 8-bit images that are processed for aesthetic beauty for the general public or multiple byte or floating point images that maintain the scientific integrity of the data for the astronomers.

To date, the following sky mosaics have been constructed using this software:

- All-sky IRAS mosaic at 1 arc minute resolution, constructed from 430 individual image patches with 4 bands each;
- 10 degree square 2MASS mosaic at 1 arc second resolution constructed from over 1000 individual images with 3 bands each;
- 2 degree square DPOSS mosaic at 1 arc second resolution, constructed from 100 individual images with 3 bands each;
- 17 degree by 14 degree DPOSS mosaic constructed from six full DPOSS plates.

In addition, the 2MASS and smaller DPOSS mosaic listed above were registered to each other and combined to produce a 6 band multi-spectral mosaic that includes both the visible and near infrared.

As mentioned above, the image mosaicking code may be run in parallel on multiprocessor systems. Each input image is assigned to a different processor and the mosaic is constructed in two stages. In the first stage, the input images are analyzed to determine latitude and longitude coverage and pixel value range. After each processor finishes analysis for the images assigned to it, the processors share their local results in a global reduction operation so that all processors have the latitude, longitude and pixel value ranges for the entire set of input images. Each processor can then fill those parts of the mosaic covered by its subset of the input images. This algorithms scales well with number of processors as illustrated in Figure 3, which shows the speedup curve on up to 64 R12000 processors of an SGI Origin 2000.

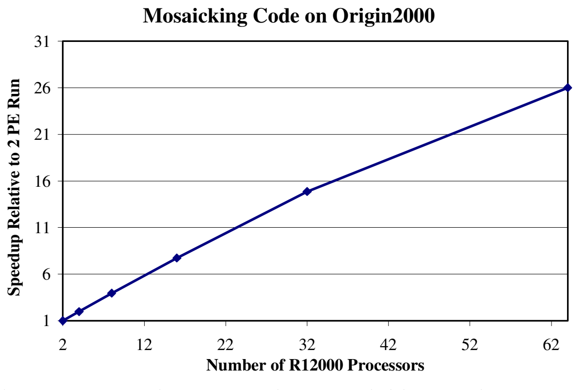


Figure 3. Speedup curve for mosaicking code on SGI Origin2000.

4. LARGE SCALE IMAGE INPUT

The internal data format used by our system is designed for high performance viewing of very large image mosaics.

4.1 I/O Framework

The SGI Image Format Library (IFL) [8] is used as the underlying framework for data input and output. The IFL provides support for opening, reading, writing and creating image files in a format independent manner. IFL includes support for the TIFF, GIF, PNG, JFIF(JPEG), SGI, PPM, Photo CD, FIT, XPM, XBM, NITF, BMP, Alias|Wavefront, and YUV file formats. This list represents most of the standard image formats used in various segments of the industry. In addition, the IFL is easily extensible, allowing users to define support for their own custom data formats. The standard formats that are supported are not suitable for very large images. Since the digital sky data sets exceed the terabyte level in size, a custom data format, described below, is necessary for high performance visualization.

4.2 Tile File

The size of a mosaic that can be constructed and viewed is limited only by the available disk space. A full sky mosaic at 1 arc second resolution in one spectral band approaches one terabyte in size. For high performance viewing of these large mosaics, the data is stored in a custom hierarchical, tiled format with two defining characteristics, illustrated in Figure 4. The data is stored as a series of 512 X 512 pixel "tiles" at multiple resolutions, including full resolution, half resolution, quarter resolution, all the way down to the resolution where the entire image fits in a single tile. The tiled nature of the data storage format permits any subset of the data to be quickly referenced and extracted for viewing without requiring that all of the data be read into memory. It also permits the image to be quickly extracted and served for viewing at any arbitrary resolution by resampling with a kernel no larger than 2 X 2 pixels.

In summary, the tiled nature of the data permits rapid panning to any arbitrary location and the hierarchical resolution storage permits rapid zooming to any arbitrary zoom level. Also, intelligent data caching keeps recently visited tiles in memory for rapid retrieval. In addition, the software includes a lookahead cache that loads additional tiles just outside the bounds of what is visible on the display and at neighboring resolution layers for improved performance. The disk storage penalty paid for this rapid panning and zooming capability is about one-third the size of the full resolution image.

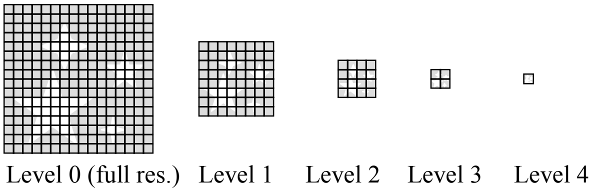


Figure 4. Hierarchical, tiled data format. Each level represents, as 512 x 512 pixel tiles, the entire image at a resolution that is half the resolution of the previous level.

4.3 Pile File

The Tile File Format described above, while providing fast access to any part of a large image at different resolution levels, has quite a few drawbacks. For example, it is impossible to extend the file by adding areas of coverage or other spectral channels without duplicating most of the data. In addition, empty areas still have to be stored on disk, thus wasting storage space. Implicit UNIX file storage holes are supported in the current implementation, but this feature vanishes when files are copied to tape or other systems from the original disk location. Another useful feature we wanted was to provide lossless and lossy data compression, a feature that forced unequal storage space requirement for tiles.

A solution to some of these problems was designed and implemented in the form of the Pile of Tiles File Format (Pile Format). The logical tile structure from the Tile Format is preserved, keeping the two formats compatible. In this format, an image stored on disk has two components, a data file and an index file. Each tile may be independently compressed, and a level of indirection is introduced between the index of a tile in the tile structure and the data storage address within the file. Within the data component the compressed tiles are concatenated, and the offset of the data for each tile, together with the size of the compressed tile is stored in the index file. As a special case, a tile offset and size of zero indicates a completely black tile, and no further data storage space is required for that tile. This provides support for explicit holes in the file structure, making this format a better match for sparse image storage.

The Pile Format files are guaranteed to be consistent during updates, even in a multiprocessor environment, since each tile is written as an atomic operation. Therefore, examination of a partially written pile file is possible at any time, providing for a quick progress check. Another unique feature of the Pile Format is that it is possible to have multiple index files pointing to the same data file, since the index file is stored in a separate file and both the data file and the index need to be specified to open a data set. A replacement tile always gets appended to the end of the data file, and then the corresponding index gets updated. If a copy of the previous index file is preserved, both the old data set and the new one can be accessed.

The Pile Format also supports image compression by applying public domain implementations of either lossless or lossy compression schemes to each individual tile. LibJPEG is used for image specific lossy compression, with each tile being stored as a monochrome jpeg image. For lossless compression, either libzip or libbzip2 are used. The file access library structure is easy to expand to provide additional compression schemes as needed.

5. LARGE SCALE IMAGE OUTPUT

Just as the image viewing software scales on the input side, the software also scales on the output side, allowing displays ranging from single screen workstation displays up to large PowerWall displays. A PowerWall display is a synchronized matrix of display screens that act together as a single large display. For example, a 3 X 2 PowerWall composed of 1280 X 1024 pixel screens provides an effective resolution of 3840 X 2048 pixels. Single pipe and multi-pipe support is provided. In the single pipe case, the hardware views the PowerWall as one large display screen so no software synchronization is necessary. However, for multi-pipe configurations, each component screen of the PowerWall is separately managed. In this case we use an implementation of the MPI (Message Passing Interface) standard [9] to synchronize the screens in software. MPI defines a collection of core routines for inter-processor communication via message passing. An important advantage to using MPI is its portability to a wide variety of distributed and shared memory multiprocessors and networks of workstations.

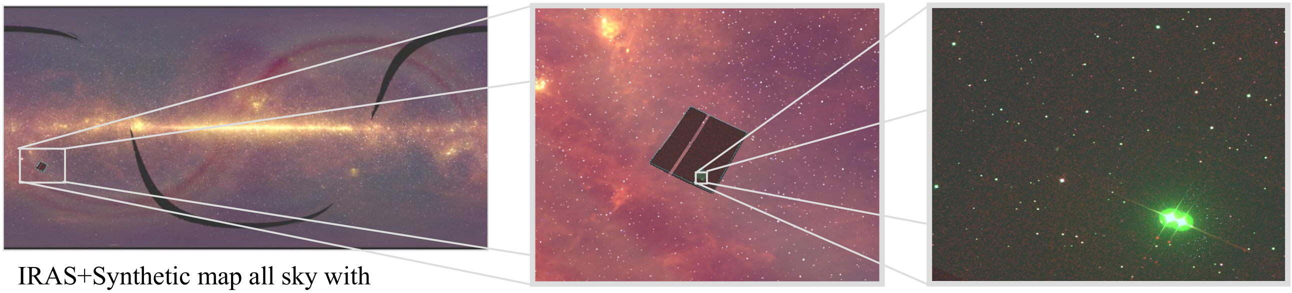
The software can be configured to use any rectangular subset of the available display screens. The supported display configurations are specified in a configuration file and are easily selectable at run-time with a command line option or by setting an environment variable. High performance on a 3 X 2 PowerWall display (3840 X 2048 effective resolution) has been demonstrated.

6. RUN TIME DATA FUSION

In this section we discuss the technical details about how multiple data sets can be automatically composited at run time. This automated compositing is a powerful visualization technique that can be used to generate composite images with greater information content than is contained in the individual components. Information about the underlying image processing framework that is used is provided below, followed by a discussion of the three compositing types that are supported.

6.1 Image Processing Framework

The SGI ImageVision Library (IL) [8] is used as the underlying image processing framework for manipulation of the input images for display. The IL provides parallel processing support, integrated memory management, flexible input/output storage options and a large pre-built set of image processing operators. Using the IL requires building an image processing chain from available and custom modules and then issuing a request for all or part of the output image. The processing engine follows the dependency chain and assigns processors to modules in the proper order to generate the requested output. For this work, the image processing chain illustrated in Figure 2 was constructed to provide support for run time data fusion. As described below, three types of data fusion are supported, spatial compositing, wavelength compositing, and image overlays.



IRAS+Synthetic map all sky with 2MASS+DPOSS high resolution inset.

Arc minute

Arc second

Figure 5. Automatic data set compositing.

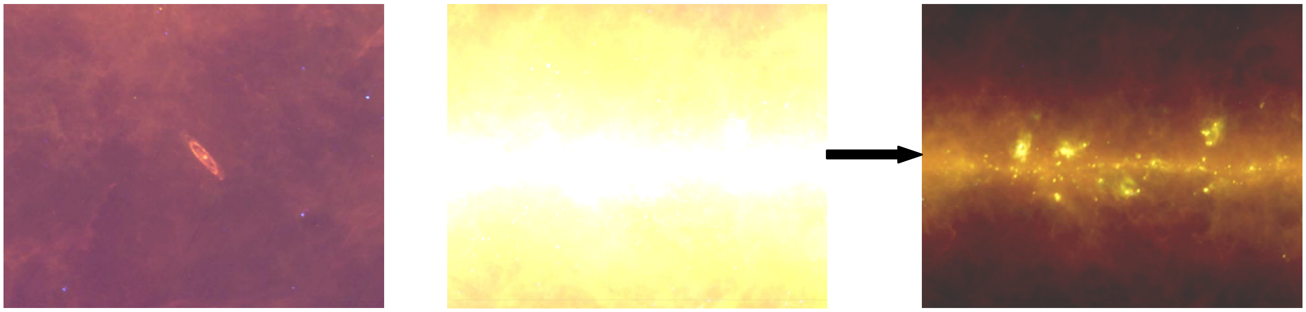


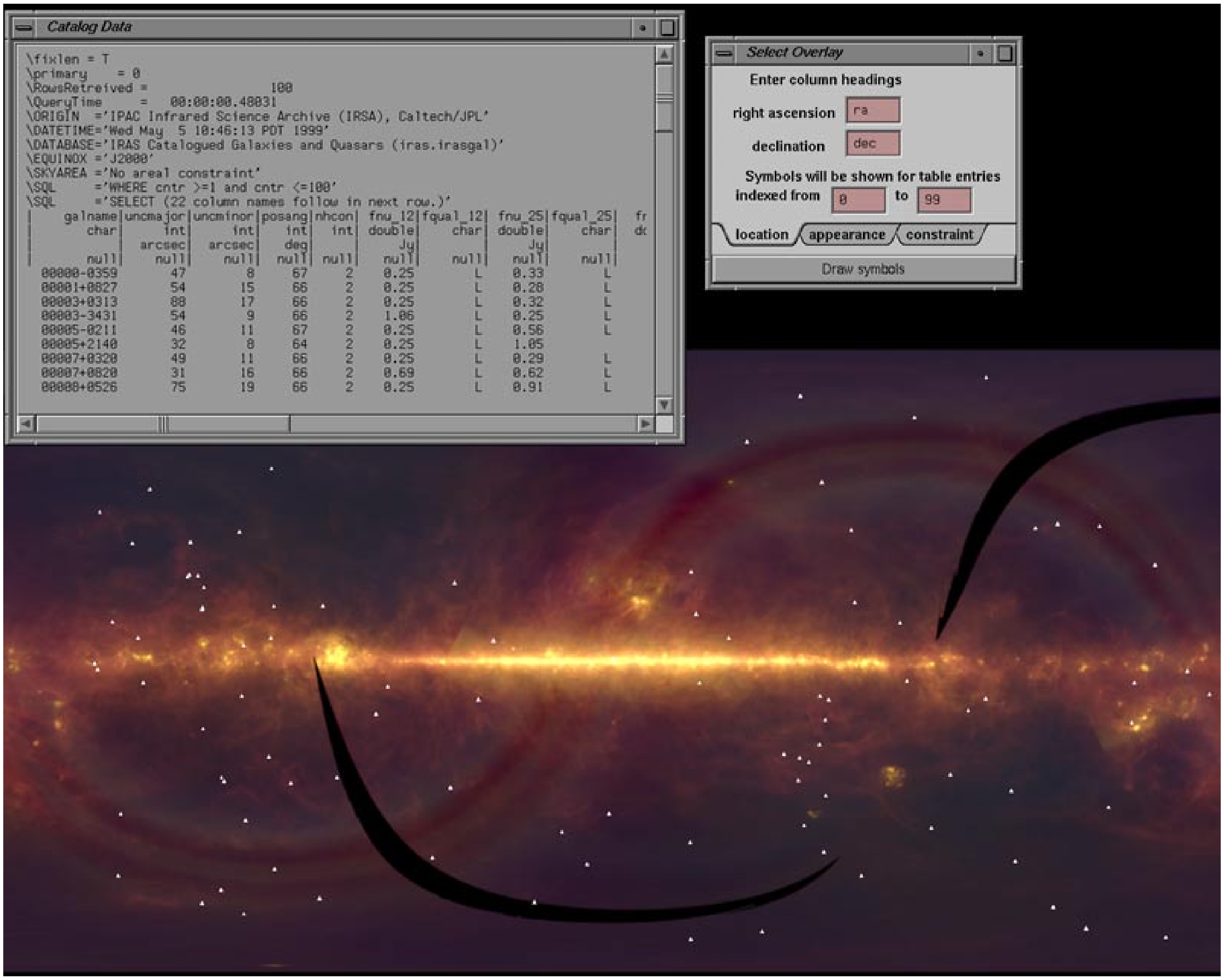
Figure 6. In the full sky IRAS image, the brightness and contrast settings for optimal viewing of the Andromeda galaxy (left) causes the center of the Milky Way to saturate (center). The brightness and contrast can then be adjusted to enhance the structure at the center of the Milky Way (right).

6.2 Spatial Compositing

In spatial compositing, multiple images having potentially different resolutions and covering different regions of the sky are combined to produce a single image. High resolution images of regions of interest are viewed as insets on top of lower resolution imagery covering much larger regions of the sky. As an example, Figure 5 shows a high resolution, multi-spectral 2MASS and DPOSS mosaic composited on the fly with an all sky IRAS mosaic. The run time compositing means that the two data sets may be stored in separate files on the disk or even on different hosts. The lower resolution IRAS data set does not have to be oversampled to match the arc second resolution of 2MASS and DPOSS. The capability of compositing at run time is a tremendous advantage since nearly a terabyte of disk storage per band would be required to store the all sky IRAS mosaic at arc second resolution. Finally, it should be noted that all the image and catalog navigation capabilities described in Section 8, including smooth pan and zoom to any part of the data, still work on the composite image.

6.3 Wavelength Compositing

In wavelength compositing, multiple co-registered images covering the same region of the sky are combined. The user can select the images to map to each of the red, green and blue video channels. This allows generation of novel, multi-spectral views of the sky. For example, the 2MASS H band in the infrared may be mapped to red and DPOSS F and J bands to blue and green, respectively. Objects that appear red would then be easily identifiable as 2MASS sources that do not appear in DPOSS. Wavelength compositing may also be used to identify quickly any registration problems between multiple data sets. For example, if we map separate data sets to different colors, small differences in position for objects in separate surveys would be identifiable at a glance. At run time, this mapping of data sets to video channels may be changed to interactively examine multiple sky surveys. Wavelength compositing is a key feature required for visualization of distributed data sets since it permits each of the images being joined to be served from different remote hosts. Also, as in spatial compositing, the viewer treats the composite image the same as any other image, permitting smooth pan and zoom.



```
\fixlen = T
\primary    = 0
\RowsRetreived =                 100
\QueryTime     =   00:00:00.48031
\ORIGIN  ='IPAC Infrared Science Archive (IRSA), Caltech/JPL'
\DATETIME='Wed May  5 10:46:13 PDT 1999'
\DATABASE='IRAS Catalogued Galaxies and Quasars (iras.irasgal)'
\EQUINOX ='J2000'
\SKYAREA ='No areal constraint'
\SQL     ='WHERE cntr >=1 and cntr <=100'
\SQL     ='SELECT (22 column names follow in next row.)'
|    galname|uncmajor|uncminor|posang|nhcon|  fnu_12|fqual_12|  fnu_25|fqual_25|   fr
|       char|     int|     int|   int|  int|  double|    char|  double|    char|   dc
|           |  arcsec|  arcsec|   deg|     |      Jy|        |      Jy|        |
|       null|    null|    null|  null| null|    null|    null|    null|    null|
  00000-0359       47        8     67     2     0.25        L     0.33        L
  00001+0827       54       15     66     2     0.25        L     0.28        L
  00003+0313       88       17     66     2     0.25        L     0.32        L
  00003-3431       54        9     66     2     1.06        L     0.25        L
  00005-0211       46       11     67     2     0.25        L     0.56        L
  00005+2140       32        8     64     2     0.25        L     1.05
  00007+0320       49       11     66     2     0.25        L     0.29        L
  00007+0820       31       16     66     2     0.69        L     0.62        L
  00008+0526       75       19     66     2     0.25        L     0.91        L
```

Enter column headings

right ascension: ra

declination: dec

Symbols will be shown for table entries indexed from 0 to 99

Figure 7. Catalog data may be viewed as ASCII text in a scrollable window or as an image overlay.

6.4 Image/Catalog Relation

The third type of data fusion that is supported is the compositing of sky images with information derived from sky catalog data. Catalog entries may be overlaid as shapes drawn on top of the images with size or color determined by the values in any catalog column. These overlays will pan and zoom in concert with the images. Furthermore, the images and catalogs are tightly coupled, allowing the user to do such things as select a region of the image and see those objects highlighted in both the image and in the catalog, as illustrated in Figure 8. Alternatively an object in the catalog may be selected to highlight it in the image or to jump to its position in the image.

7. REMOTE ACCESS

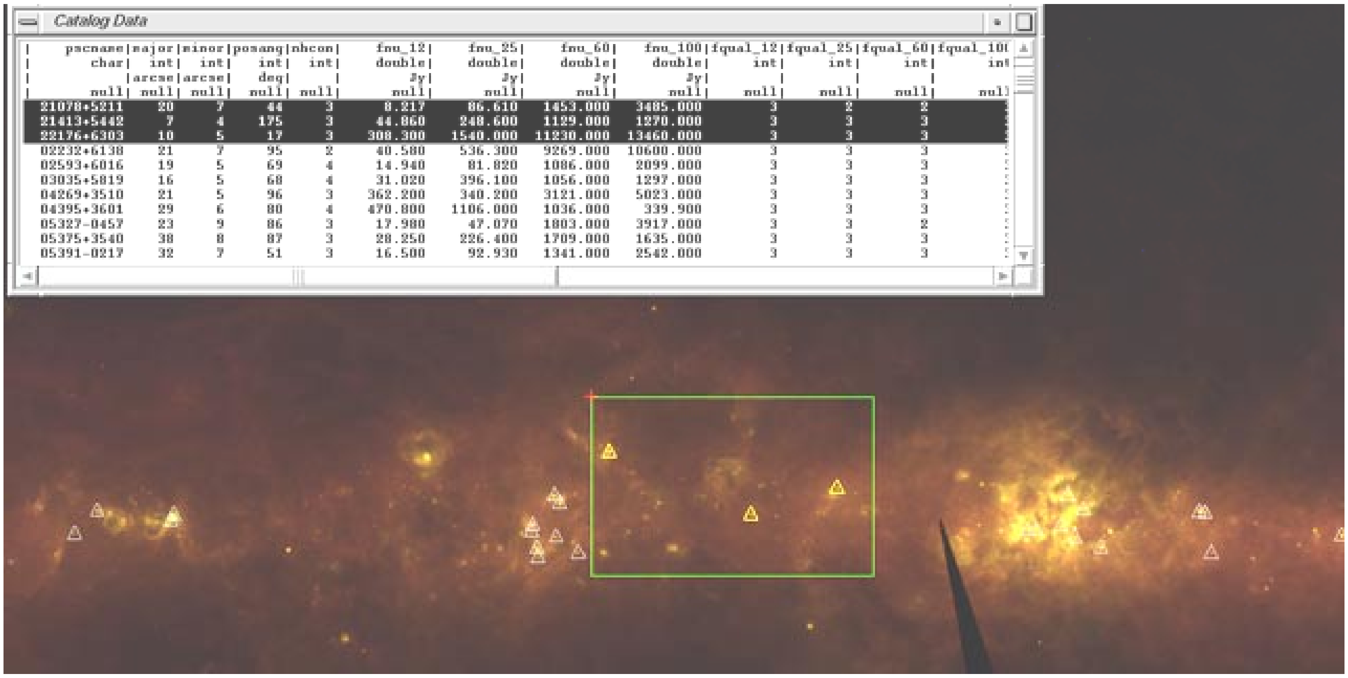
When the viewer needs to access data that resides on a remote host, the network image server must be used. We expanded the IFL (described in Section 4) by adding a network file format, allowing for image access from any networked computer. A client-server strategy was employed, in which the image server uses the IFL to open local image files and serves them to the client via a TCP socket. An IFL image loader was added to the client viewer, permitting it to access any IFL supported image residing on the server machine using the network file format. Since a network image file in this implementation is no different from any other IFL image file the servers can be cascaded, with various processing being done by intermediate sites. When used in conjunction with the Pile or Tile formats, the protocol can be used to fetch each tile data independently, with possible decompression at either the server side or the client side.

8. IMAGE AND CATALOG NAVIGATION

The software that has been developed permits high performance visualization of both sky image and catalog data. Several features are provided to enable efficient navigation of the potentially huge images. Mouse and keyboard interfaces for smooth, variable speed panning and zooming are provided. A Global Map View of the data set is a popup window that shows the entire data set with a box highlighting the region that is currently visible on the display. Users may jump to any location in the image by clicking at the location in the Global Map View. Another way to quickly jump to any location in the image is to specify that location in either pixel or sky coordinates in the Coordinate Selection window. Galactic, ecliptic, and celestial coordinate systems are supported. The Coordinate Selection window may also be used to retrieve the pixel or sky coordinate of any pixel that is selected by clicking with the mouse on the image.

The image mosaicking software described in Section 3 is capable of constructing mosaics having many spectral bands. The image viewing software permits any subset of three bands to be mapped to red, green, and blue at run-time. This allows the user to do such things as view both visible and infrared bands from different sky surveys simultaneously and then switch back to all visible or all infrared.

Keyboard and graphical user interfaces for brightness and contrast adjustments are provided, and may be applied to all three video channels (red, green, and blue), or to any channel individually. For example, this feature allows viewing of both Andromeda and the center of the Milky Way in our IRAS mosaic, as illustrated in Figure 6.



```
|     pscname|major|minor|posang|nhcon|     fnu_12|     fnu_25|     fnu_60|    fnu_100|fqual_12|fqual_25|fqual_60|fqual_10
|        char|  int|  int|   int|  int|     double|     double|     double|     double|     int|     int|     int|     in
|            |arcse|arcse|   deg|     |         Jy|         Jy|         Jy|         Jy|        |        |        |
|        null| null| null|  null| null|       null|       null|       null|       null|    null|    null|    null|    nul
  21078+5211    20     7     44     3      8.217     86.610   1453.000   3485.000        3        2        2
  21413+5442     7     4    175     3     44.860    248.600   1129.000   1270.000        3        3        3
  22176+6303    10     5     17     3    308.300   1540.000  11230.000  13460.000        3        3        3
  02232+6138    21     7     95     2     40.580    536.300   9269.000  10600.000        3        3        3
  02593+6016    19     5     69     4     14.940     81.820   1086.000   2099.000        3        3        3
  03035+5819    16     5     68     4     31.020    396.100   1056.000   1297.000        3        3        3
  04269+3510    21     5     96     3    362.200    340.200   3121.000   5023.000        3        3        3
  04395+3601    29     6     80     4    470.800   1106.000   1036.000    339.900        3        3        3
  05327-0457    23     9     86     3     17.980     47.070   1803.000   3917.000        3        3        2
  05375+3540    38     8     87     3     28.250    226.400   1709.000   1635.000        3        3        3
  05391-0217    32     7     51     3     16.500     92.930   1341.000   2542.000        3        3        3
```

Figure 8. Image to catalog relation.

The catalogs may be viewed in ASCII text in a scrollable window and as image overlays, as illustrated in Figure 7, or as synthetic maps. With overlays, which are vectors drawn over the image, the shape, size, and color of the overlay objects may be set according to the values in one or more columns of the catalog. With synthetic maps, the catalog positions and values are used to generate a pixelated image. For instance, the Hipparcos catalog was used to generate a synthetic map of the sky with 118,218 stars down to magnitude 14.

The image and catalog viewing capabilities are tightly coupled, allowing easy relation of a location in the images to catalog entries for celestial objects in the proximity and vice versa. For instance, the user may select a region of the sky in an image and see the catalog entries for those objects in that region highlighted in both the image and in the catalog window, as illustrated in Figure 8. Alternatively, the user may select a catalog entry in the scrollable list and see that object highlighted in the image or jump to the position of that object in the image. This demonstrates both image to catalog and catalog to image relations.

9. SUMMARY

In this paper we presented our system for visualization of large images and sky catalogs. The system consists of software for construction of large image mosaics, a network image server that can deliver image tiles from remote hosts, and a high performance viewer. The viewer has support for smooth pan and zoom of large images, automatic insetting of high resolution images, automatic joining of multi-wavelength images, and image to catalog relation.

ACKNOWLEDGMENTS

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

[1] T. Boroson, R. Brunner, D. De Young, S. Djorgovski, R. Hanisch, S. Strom, and D. Tody, 2000, White Paper, Toward a National Virtual Observatory: Science Goals, Technical Challenges, and Implementation Plan, http://astro.caltech.edu/nvoconf/white_paper.pdf.

[2] http://www.ipac.caltech.edu/2mass.

[3] http://astro.caltech.edu/~rrg/science/dposs_public.html.

[4] http://www.lcse.umn.edu/research/powerwall/powerwall.html.

[5] Thomas A. McGlynn and Keith A. Scollick, A User's Guide to SkyView, Version 3.0, January 14, 1997, http://skyview.gsfc.nasa.gov.

[6] NASA/GSFC Astrophysics Data Facility, A User's Guide for the Flexible Image Transport System (FITS), Version 4.0, April 14, 1997.

[7] E. W. Greisen, and M. Calabretta, 1999, Representations of world coordinates in FITS, submitted to AAP, http://www.atnf.csiro.au/computing/software/wcslib.html.

[8] G. Eckel, J. Neider and E. Bassler, ImageVision Library Programming Guide, 1996, http://www.sgi.com/software/imagevision.

[9] Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, June 12, 1995, http://www-unix.mcs.anl.gov/mpi.

Joseph Jacob *is a Member of the Information Systems and Computer Science Senior Staff at the Jet Propulsion Laboratory, California Institute of Technology. He has been working with the Advanced Laboratory for Parallel and High Performance Applications at JPL since joining the laboratory in 1996. His research interests are in the areas of parallel and distributed computing, image processing and scientific visualization. His graduate work was completed at Cornell University where he earned the Ph.D. and M.S. degrees in Computer Engineering. In addition, he has a B.S. degree in Electrical and Computer Engineering from the State University of New York at Buffalo.*

Lucian Plesea *is a Member of the Information Systems and Computer Science Senior Staff at the Jet Propulsion Laboratory, California Institute of Technology. He has been working with the Advanced Laboratory for Parallel and High Performance Applications at JPL since joining the laboratory in 1995. Previously he was a member of the Controls Group for the Superconducting Super Collider in Dallas, Texas. His current research interests are in the areas of high performance distributed computing and visualization, with emphasis on global GIS systems. He received his MS in Electrical Engineering from the Polytechnical Institute of Bucharest, Romania.*