December 23, 1991

AIPS++ User Specifications: An Initial NRAO-Oriented Version

R.M.Hjellming, A.H.Bridle, R.J.Maddalena, D.O.S.Wood, J.A. Zensus

National Radio Astronomy Observatory[1]
Socorro, NM 87801-0387
Green Bank, WV 24944-0002
Charlottesville, VA 22903-2475

D.J.Westpfahl

New Mexico Institute of Mining and Technology and NRAO, Socorro, NM 87801

1 Introduction

AIPS++ is a software system being developed by a consortium of radio-astronomy oriented institutions in the United States and around the world. This document presents an initial NRAO-oriented view of goals and desirable capabilities of the new package. While we expect that AIPS++ will be multi-faceted, its main role should be to turn astronomical data into scientific results. It must therefore reduce telescope data into astrophysically meaningful data forms such as images, time-series and spectra, and must also aid in the astrophysical interpretation of these forms by comparison and by modeling.

Many ideas in this document were generated from discussions at a weekly *AIPS++ Development Seminar* at the Array Operations Center that began in late August 1991. In the area of single dish software, working documents produced during the preparation of *The Requirements for Data Analysis Software for the Green Bank Telescope* (Foster *et al.* 1991) were utilized. There has not, however, been enough time to involve other NRAO sites or non-NRAO astronomers in writing this document to the degree that would have been ideal. This document should therefore be viewed as an NRAO-oriented *first draft* for consideration by the AIPS++ design and development group. We expect that it will be significantly modified as the AIPS++ design progresses, and hope that a wide spectrum of inputs from the potential user community will be part of that process.

The AIPS++ project itself stems partly from recommendations made by the Software Advisory Committee (SWAG), which made its final report (Cornwell 1990) to the NRAO Director in September 1990. The SWAG report itself addressed user specifications for a new software system, many of which parallel those made here.

2 Specifications and Object-Oriented Analysis

The use of written specifications for software systems has not been generally successful. In traditional software development this is in part because everything from user interface to implementation details is prescribed by people who will not be responsible for implementing the software. The object-oriented approach planned for AIPS++ requires high level specifications as a prelude to object-oriented analysis (OOA), which itself is a prelude to object-oriented programming (OOP) in which classes, etc. are designed, prototyped, and implemented. For

[1] The National Radio Astronomy Observatory is operated by Associated Universities, Inc. under a cooperative agreement with the National Science Foundation.

this approach we have assumed that specifications should provide a complete set of material for analysis. None of us are (as yet) familiar with OOA and OOP, so we have tried to avoid constraining our thinking as users.

3 Guiding Principles

These specifications describe the capabilities needed in AIPS++ by astronomers who use NRAO telescopes. We attempt to avoid expressing opinions on how such capabilities should be implemented in AIPS++. However, because AIPS++ should be optimized for the astronomer user, we do specify some aspects of the user interface that we consider essential.

AIPS++ must anticipate a wide range of experience within its user community. Both the user interface and the off-line documentation must address the disparate needs of novice (or occasional) users and of experienced users who may be analyzing technically demanding observations. To match the needs of users with a wide range of experience, a hierarchy of interfaces and documentation will be essential. Users will also need a hierarchy of programmability. At the lowest level of experience, this should allow them to connect major (and sometimes repetitive) steps in data processing conveniently. At the highest level, an efficient interface is needed to encourage development of new, experimental, algorithms and processing techniques.

The following principles are important in the design and implementation of AIPS++:

- Accountability - Data should have associated telescope performance ("monitor") and processing histories so their origins and evolution can be easily reviewed and understood by astronomers.
- Astronomical terminology and concepts - Names and labels in the data processing system should use the common language of astronomy and mathematics.
- Programmability - It should be easy for users to prepare data processing "scripts" for repetitive or multi-stage processing, and to augment the system easily with new data processing algorithms.
- Easy Customization - The user should be able to flexibly select data processing packages to be used, the style of user interface, and environmental parameters such as directory names and output devices.
- Hiding complexity - Where possible the complexity of algorithms or multi-step processing should be hidden from the novice user.
- Confidence in results - Data processing diagnostics and the capability to un-do and re-do steps in the processing are essential for telescope data so the user can understand and have confidence in the data at any processing stage.
- Range of processing styles - the same software should be usable for both post-observing processing and remote or local on-line data analysis

4 Scientific Goals

AIPS has been an acronym for "Astronomical Image Processing System". Its capabilities and users' requirements have evolved far beyond image plane processing, however. AIPS should now be seen as a general tool for turning telescope data (or theoretical calculations) into scientific results. In some cases, e.g., graphics and tables, the results should be in publishable form. Most n-dimensional images are produced only as an intermediate step between raw data and useful results, however some are constitute final scientific results and require reproduction in publishable form. A similar range of purposes has evolved for single dish data in systems such as UNIPOPS. The concept for AIPS++ should be that of an Astronomical Information Processing System.

In specifying a new software system, it is useful to consider what aspects of astronomical data processing have remained stable over the last 15 or so years. The most stable parts of array and single-dish processing systems are the *fundamental* descriptions of telescope data. For the VLA, the basic description has accumulated more attributes (e.g. spectral channels, IFs) but it is still fundamentally a visibility data set – samples of a spatial coherence function in some convenient spatial or temporal order. Similarly, a final image is still an array of calibrated pixel intensities in a known coordinate system, polarization, and observing frequency. The "end results" are scientifically meaningful quantities extracted from one or more such images. Key, and probably stable, basic ingredients of a user specification are therefore the types of *data* to be handled (e.g. visibility data, single dish spectra, images, image cubes).

Basic operations on parts of data sets, such as Fourier transforms, least squares fitting algorithms, plotting, display, are also relatively stable. We will call these basic operations *tools*. The second ingredient of a user specification is an itemized tool kit of basic operations from which more complex astronomical applications can be assembled.

In contrast, the algorithms used to calibrate, construct and interpret data sets and images evolve as the astronomical community acquires experience and sophistication in data and image analysis techniques. The algorithms are the least stable elements of present software. They continually evolve or are replaced (either as explicit programs or as informal procedures that may involve astronomer interaction). The algorithms are embodied in *tasks* which can be implemented either as specific programs in a language such as C++ or as scripts in a higher level language. Many of the tasks that are now part of the lexicon of astronomical image processing will be embodied in AIPS++ at an early stage. The tools in the kit provided by AIPS++ must however be easily usable by astronomers to carry out new tasks whose nature and scope may evolve rapidly with time.

In these terms, the core of AIPS++ must provide a generic toolbox operating on specific data types. Given the finite resources available, the limitations of AIPS++ should be more in the diversity of data that can be handled rather than in what can be done to these data.

What does the AIPS++ user specification consist of? The framework of our description is (*data + tools + astronomer = tasks*). We now fill out what we mean by *data, tools, tasks*, and the role of the *astronomer* in the processing.

4.1 The General Nature of the Data for AIPS++

The data will come principally from radio telescopes although AIPS++ must allow import of images and data from other wavelengths. The primary data types that are needed to support the VLA, the VLBA, the future mmA, and the various instrument packages on the GBT, the 12m and the 43m are as follows:

1. Telescope status information

2. Total power and phased array data sequences reflecting switched or time series observations

3. Spectra
4. Images
   (a) Planar images at radio, optical, X-ray, etc. wavelengths
   (b) Spectral cubes - images in multi-spectral regions
   (c) Time cubes - time-ordered images of variable sources
5. Coherence function (visibility) data from correlation arrays
   (a) arrays with real-time delay and phase variation correction
   (b) tape recording arrays where the correlator output is coherence function data for a range of time lags (or transformed frequencies)
6. Calibration tables
7. Data editing information
8. Computed models
9. Processing histories

Some of these data categories are naturally associated with each other; it is also important to be able to group some together when appropriate, e.g. in mosaicing observations. Some of these data types are either super-sets or sub-sets of others; it is important to be able to compose super-sets out of sub-sets and to decompose super-sets into their sub-sets.

It is important that the astronomer have access at all stages of data processing to the conditions under which an observation was made, and to what has been done to it in the data processing. The ability to wipe the slate clean has proven its utility over and over again in the VLA DEC-10 software and in AIPS. Hence the database should carry both telescope-provided status information and a processing history in formats that make it easy to "start over" if processing goes awry. This supplementary information begins with data structures with telescope information as a function of time, position, or other data identifiers such as telescope name, latitude, etc. It continues with data processing history sufficient to understand, un-do, and re-do that processing.

Another view of the data relates to different uses and time scales of use. These uses lead to three major categories of software: on-line data analysis; system support software for NRAO-staff operating and diagnosing the operation of the instrument; and observers analysis software. For single telescopes the observer has often done a major fraction of data analysis at the telescope as part of observing process. This capability was available for knowledgeable users during the construction years of the VLA, but was terminated when on-line and off-line resources became to strained too support this "luxury". Recent hardware and networking developments have made such on-line data analysis feasible even for high data rate instruments like the VLA and single dishes with fast sampling spectral processors. Currently under way at the VLA is a project to once again do nearly real-time data processing, using fast networking and powerful workstations.

It is planned to operate the future mmA (Millimeter Array) with full capability for local and remote data analysis on-line. The GBT will have full remote and local analysis capability. In all these system data should be accessible to the user as soon as practicable in nearly real-time. For all these reasons data analysis software in the AIPS++ system should provide for the needs of the above mentioned three categories of software.

4.2 Tools vs. Tasks

Each type of data will have its own set of tools, but many tools will be generic. One-dimensional data processing tools apply generally to single dish spectra, time series data, and slices through n-dimensional data. There will also be tools to transform one type of data to another: e.g., visibility data to "dirty" images. Tasks such as the various mosaicing algorithms can go between images and many other types of data. The line between tools and tasks is fairly arbitrary and today's task may become tomorrow's tool.

Tools are the distillation of tasks into basic, reusable elements. Tools turn initial data elements into modified or different data elements. Tasks embody astronomical goals, and may sequentially use tools to obtain the data needed for specific scientific purposes. Tools transform things with few I/O options, whereas tasks are built from sequences of tools and are flexible with a variety of I/O options. The basic tool kit should have the following organizational categories:

1. Display/plotting
2. Querying and/or summarizing observations and specified data (e.g., give me a table of T_{sys}, Elevation, Time, Barometric Pressure, Dew point for the last two days)
3. Input/Output in a variety of forms e.g. FITS, ASCII, improved FITS(?)
4. Editing/Correction
5. Calibration
6. Combination/Concatenation
7. Condensation/Selection/Averaging
8. Statistics
9. Smoothing/Filtering
10. Mathematical functions for numbers, arrays, etc.
11. Spreadsheet-like processing with arrays and numbers
12. Geometric corrections
13. Simulation
14. Fitting
15. Data base manager
16. User interface
17. System interface for file systems and host operating system
18. Help and documentation system

This description is useful for organizational purposes, but we note that it can also provide a way to reduce the apparent complexity of data reduction. In a menu-based user-interface, categories such as these can form the highest level in a list of options. Note that we do not expect that the tools will be the same for each type of data, but that tools appropriate for each type of data will be provided in each relevant category.

It is important that AIPS++ should provide sophisticated users with efficient ways to assemble new tasks from the basic tool kit. A problem with existing software is that tasks are built as monoliths, from which it is difficult to extract the powerful generic tools for use in other contexts. AIPS++ must "retain easy access to the tool kit" for the sophisticated user, and not constrain task development to simply cloning or modifying existing tasks.

The following section specifies lists of tasks for which AIPS++ should initially provide tools. Then we address a partial list of tools. Identifying which tools are necessary for which tasks is an important goal. Then we will describe in more detail the data elements that AIPS++ should be able to process, processing problems that require special treatment, the question of priorities, and the future development of specifications.

5 List of Tasks

We have described AIPS++ as being based around data and methods to manipulate those data, and we have emphasized that initially the algorithms/tasks to do specific things are less important. Nevertheless we want to include a list of specific things that the system should be able to do. This should be viewed as a *checklist*. We want to be able to do the following things eventually, but not all of these need be present initially. These things should be possible in AIPS++ via fairly high-level scripts or sequences of operations. The following list of tasks contains mixtures of things that are well understood and commonly used, and techniques that do not yet exist in working software.

5.1 General Attributes of AIPS++

There are a number of things about the package itself which can be specified independently of the hierarchy of data/tools/tasks.

1. User selection of "packages" to be used, e.g. single dish oriented setups, aperture synthesis oriented setups, VLBI oriented setups, etc.
2. On-line, hierarchical (hyper text) documentation: including parameter help (functions, units, glossary), syntax/grammar checking before execution, algorithm descriptions, listing commands for specific topics, comment reporting facility, what-is-new listings
3. Context sensitive help that leads the user to the appropriate "help" information when an error occurs
4. Warning of high resource usage (CPU, disk, tape) before programs go into execution (advise user and request confirmation)
5. Processing history for documentation and re-processing (executable history format is desirable)
6. Astronomical and algorithmic terminology as much as possible

7. Command line language interface capability

8. Multi-window input and output capability

9. Selection of user interface

10. Selectable input and output data streams and display devices

11. Access to both data values (images, spectra, data subsets) and "header" values and a good sampling of physical and mathematical constants

12. Mathematical operations and functions for numbers, vectors, and arrays allowing data transformation

13. Logic operators; WHILE-DO, DO-UNTIL, IF-THEN-ELSE and other control (FOR LOOPS, CASE Statements) programming capabilities

14. String manipulation facility

15. User selection of external editor for files and command lines

16. Review, editing, and re-execution of sequences of AIPS++ command language statements and command lines

17. Storage and retrieval of "program" parameter environment

18. Access to operating system commands from inside AIPS++

19. A flexible and complete system for file access/listing that include sub-directories, wild card searching and listing, and information of file sizes, modifications dates, etc.

5.2 Obtain and Process Data from Specific Instruments

5.2.1 Involvement in Observing Process

1. Preparing instrumental control information

2. Simulate results of instrumental observing

3. See instrumental status information

4. See instrumental output in meaningful form during observing process

5. Automatic first order data editing and calibration whenever possible

6. As much immediate data processing and display as feasible in real-time

7. Feedback to change observing program

8. Telescope/array data written on transportable storage media for both permanent storage (telescope/array data archive) and subsequent processing/re-processing

9. Summarize, copy, average, select telescope archive data

10. Building data super-sets out of simple data sets; reducing data sets into sub-sets

5.2.2 Single Telescope and Summed/Phased Array Data Processing

1. Telescope archive data translation from storage media to disk data base
2. Data editing
3. Standard and user definable data correction for known instrumental defects
4. Standard and user definable data calibration
5. Spectral manipulation and display (see under "1-d processing/analysis")
6. Image computation (basket-weaving) from total power data
7. Flexible spectral fitting to baseline and components; interpolation through bad channels
8. Statistical software (for array fitting) with error analysis; non-linear fitting with error minimization
9. Pulsar data collection and processing
10. Telescope pointing and beam pattern data analysis
11. Analyzing telescope performance data: pointing, calibration, on-off, telescope-tipping, focusing, and holographic data
12. Polarization data reduction
13. Import/export single dish FITS and other formats
14. Fitting, filtering, convolving, FFT and IFFT of data arrays
15. Deconvolution of 'channel' shapes from 1-, 2-, and 3-D data; deconvolution of 'frequency-switched' data.
16. Data handling for fast sampling spectrometers with from 128 to 32768 channels of data, producing very large data "cubes"

5.2.3 Array Data Processing - Instrumental Coherence Function Data Sets

1. Telescope and array archive data translation from storage media to disk data base
2. Use of instrumental status data for data editing and evaluation
3. Application and de-application of astrometric correction factors
4. Pointing, baseline, and beam pattern determination and analysis
5. Selective data listings in column and matrix form
6. Phased summing of spectral channels for reference/shifted positions
7. Flexible display of coherence data in many forms of 2-D plots and 3-D t-Baseline-Intensity or u-v-Intensity images
8. u-v plane zonal averaging and display

9. Command line editing specification capabilities

10. Interactive editing with t-Baseline and u-v ellipse displays of amplitude

11. Solve for, display, and analyze instrumental calibration parameters (amplitude and phase vs. time; spectral response functions; instrumental polarization parameters, absolute polarization position angle) with storage in calibration data base

12. Flexible interpolation and averaging of calibration data as part of reversible data calibration process

13. Selection, copying, and averaging of coherence data sets

14. Copying complete or selected instrumental data sets and associated calibration tables to storage media

15. Restoration of instrumental data sets and associated calibration tables from storage media to disk

16. "Build-up" imaging in nearly real-time: summed radial transforms for linear arrays; regridding and imaging of incrementing data sets for non-linear arrays; and direct transform imaging of incrementing data sets

17. Translation of instrumental coherence data sets to data sets which can be merged with those of other instruments

5.2.4 Array Data Processing - VLBI Correlator Data Sets

VLBI correlator data require more complicated astrometric and geodetic models than those for phase-linked arrays, in addition to requiring VLBI-specific processing tasks.

1. Basic Calibration

    (a) Data consolidation

        i. Sort data from different VLBI correlators
        ii. Merge data sets from different VLBI correlators
        iii. Merge tables of information from different VLBI correlators

    (b) Amplitude calibration

        i. Continuum calibration from application of T_{sys}, T_{ant}, and gain curves for each antenna
        ii. Spectral line calibration using gains from auto-correlation (self-spectra) data
        iii. Determine and apply complex band pass correction to all types VLBI data

    (c) Fringe-fitting

        i. Determine residual fringe rates and delays simultaneously - for single or multiple frequencies (global fringe fitting)
        ii. Orbiting VLBI antenna data may require additional determination of delay acceleration
        iii. Orbiting VLBI antennas may need fitting of data for multiple fields because the field of view is very small ($\sim$ milliarcseconds for VSOP)

(d) Editing
    i. Need fast, graphical baseline-by-baseline editor
    ii. Editing based on closure phases
    iii. Improved, larger data volume displays coupled with interactive editing
    iv. Interactive editing and smoothing of calibration data

2. Astrometric, Geodesy, and Phase Referencing processing
    (a) Require full accountability, including model of the observations, correlator model, instrumental behavior data, and full geometry model for all antennas (the "total observables")
    (b) Require global, accurate definition and implementation of time
    (c) Need precise reference frame defined by geodetic data
    (d) Need Earth orientation parameters, UT1, polar motion, etc.
    (e) Need full antenna models (axis offsets etc., alt-az etc., equat etc.)
    (f) Must be possible to un-do the correlator model, apply a new model, and regenerate fringe rate residuals

3. Post-fringe fitting applications
    (a) Continuum imaging, combining spectral line channels
    (b) Spectral line imaging, with ability to perform spectral phase-referencing
    (c) Fringe-rate imaging
    (d) Multi-frequency synthesis
    (e) Polarization data processing - requires rigorous treatment involving different properties for each antenna (parallactic angle, polarization modes are different)

4. Data Format Requirements
    (a) Handle data from VLBA and other correlators
    (b) Multi-spectral channel, multi-IF, multi-polarization visibility data handling
    (c) Flexibility in data format definition – different numbers of channels in different IF's, different channel bandwidths
    (d) Visibility data in complex representation with associated, precisely-defined errors as a function of spectral channel and time
    (e) Ability to store different frequencies simultaneously, with capability to apply fringe rate residual solutions from one band to another
    (f) Storage of auto correlation data in the same data base
    (g) Organization of data in sub-arrays, antenna-dependent parallactic angles and horizons
    (h) Ability to interactively change ancillary correction and calibration "data"
    (i) Import and export data from/to other VLBI packages (e.g. CIT)

5. Imaging

(a) Triple-correlation imaging (S/N limitations) – summing closure phases in suitable combinations to eliminate instrumental phase errors

(b) Coherence analysis

(c) Space VLBI moving baselines

(d) Near field imaging (Sun, planets, planetary probes)

(e) Model fitting

5.3 Import and Export Data from/to Different Instruments and Models

Catalogs and Archives

1. Search astronomical catalogs
2. Search instrumental data catalogs
3. Select data from instrumental data catalogs
4. Extract spectra, images, and point source information from instrumental data catalogs
5. Import spectra, images, and point source information with transformation to standard formats

5.4 Interfaces to Other Processing Software

It is important that users be able to exchange the main data forms freely with other processing and analysis packages, such as IDL/PV-Wave, AVS, IRAF, NOD2, ANALYZ, UNIPOPS, CLASS, DAOPHOT, COMB, SPA, MIDAS, VISTA, Mathematica, MathCAD, theoretical modeling programs, etc. with and without necessarily writing data tapes. FITS disk and ASCII table formats will address many, but not all, such data transfers. In some cases translation will be necessary for input to specific package formats. The user community should be surveyed for its preferences on target packages and transfer formats to see if any are so popular that the consortium, rather than a specific user group, should write the export/import codes.

1. Export/import one-dimensional data streams in FITS format, tabular arrays, etc.
2. Export/import images in FITS format, tabular arrays, etc.
3. Export/import cubes in FITS format and other self-descriptive formats
4. Export/import coherence function data sets in FITS format, tabular arrays, etc.
5. Import tabular data from user/model generated files obtained from theoretical calculations
6. Translate data arrays to and from FITS and other image formats

5.5 Single Telescope and Summed/Phased Array Processing

1. Continuum data analysis and imaging
   (a) Multi-beam image reconstruction
2. Spectral line processing and analysis (see also 1-d processing/analysis)
   (a) Channel combination and weighting to determine "continuum" image
   (b) Continuum subtraction from line coherence data sets
   (c) Multi-beam spectral line cube imaging
   (d) Total power imaging of segments of a mosaic image with transformation to visibilities to merge coherence function data
3. Time series processing
   (a) Power spectrum analysis (see also 1-d processing/analysis)
   (b) Period phasing, stacking
   (c) De-dispersing
   (d) Period searching
   (e) Drift rate analysis

5.6 Stationary Coherence Function Imaging

1. Imaging after subtraction for sources in the coherence function data
2. Continuum subtraction from line coherence data sets
3. Weighted, gridded, tapered u-v plane (dirty) imaging and point-spread-function (PSF) computation
4. Estimation and input of zero-spacing flux density and appropriate weighting from visibility data.
5. Direct (non-FFT) transform image computation
6. De-convolution from dirty image and PSF: Högbom CLEAN, Clark-Högbom CLEAN, Cotton-Schwab CLEAN, smoothness-stabilized CLEANs, maximum entropy, maximum emptiness.
7. Multiple subfield imaging with ungridded subtraction (MX)
8. Multi-frequency imaging
9. Three-dimensional FFT imaging (wide-field problem)
10. Near-field imaging of nearby comets and asteroids

5.7 Combined Total Power and Coherence Function Imaging

1. Sensible display of mosaic-oriented visibility data; individual pointings or multiple pointings.
2. Calibration of mosaic databases.
3. Self calibration and editing of all pointings in one processing step. Automated (black box) self calibration and editing.
4. Linear combination of pre-deconvolved images, weighting determined by the primary beam.
5. Linear mosaic algorithm/linear deconvolution (MOSLIN in SDE).
6. Nonlinear (MEM based) mosaic algorithm (VTESS, UTESS in AIPS, MOSAIC in SDE).
7. Ability to easily manipulate and image many (1024) spectral line channels in mosaic databases.
8. Continuum subtraction and line subtraction from mosaic visibility databases.
9. Software to determine the primary beam (PB) from a mosaic image and the mosaic database (PB self calibration).
10. Ability to deal with many primary beams in different forms:
    (a) analytic PB types (mmA, VLA)
    (b) 1-D arrays fit to a rotationally symmetric PB
    (c) 2-D non rotationally symmetric beams; must rotate on the sky as parallactic angle changes.
    (d) Flexibility for user modification of the PB models.
11. Pointing self calibration to determine corrections for both single dish and visibility data
12. Removal of point sources from mosaic database.
13. 3-D mosaicing allowing for non coplanar baselines
14. All mosaicing tasks must be able to deal with (calibrate, image) beam switched total power.
15. Cross calibration (enforced consistency) between data taken with different instruments (flux scale, pointing).
16. The synthesized beam must be monitored across the mosaic image. Flagging or reweighting of some data on some pointings may be required to make the fit beam reasonably constant across the image.
17. Simulation program for mosaic databases, including error generation: thermal noise, pointing errors, primary beam errors, atmosphere, surface errors, beam switching for total power, etc.

5.8 Model-dependent Modification of Coherence Data and Images

1. Source subtraction in u-v domain
2. Correction of data for source motion - asteroids, comets
3. Solar data
4. Planetary data - removal of effects of disk emission, motion, and rotation
5. Highly variable sources in otherwise stationary field

5.9 n-Dimensional Processing and Analysis

5.9.1 One-Dimensional Tasks

One-dimensional data sets will include line spectra at single sky positions, time series of single parameters, or "slices" across continuum images. These have many processing requirements in common and so are treated together here. We describe them all here as "profiles," – the axis may be frequency/velocity, temporal or spatial in different applications. Some are generalizable to 2- or 3- dimensions.

1. Polynomial baseline fitting and subtraction
2. Fitting of sinusoidal and other simple spectral baseline shapes
3. Multiple component fitting with residual displays (Gaussians, parabolas, etc.)
4. Multiple profile superposition
5. Profile smoothing and filtering
6. Profile stacking/averaging
7. Re-gridding to change resolution and center
8. Numerical differentiation
9. clipping and other filtering (impulse, etc.) operations in data and data transform planes
10. Multi-panel profile plotting (tiling)
11. Profiles for selected positions superimposed on (or plotted with lines to position in) images
12. Power spectrum analysis
13. General calculus to replace any array by a function of itself and constants/header values

5.9.2 Two-Dimensional Tasks

Two-dimensional data sets are any functions of two co-ordinates that make up a meaningful array for astronomical users. We describe them all as "images" whether or not the co-ordinates are both spatial. Note that "images" may be assembled from "profiles" (see above) or by dissecting "cubes" (see below). AIPS++ must contain generalized facilities for such assembly and dissection.

1. Image calculus - deriving images from other images using general mathematical syntax
2. Polarization imaging (polarimetric algorithms for image combination)
3. Source recognition and classification – reducing image to a list of sources with positions and two-dimensional Gaussian fits
4. Isolated source model fitting with multiple Gaussian components
5. Interactive input of rectangular and curvilinear boundaries for image arithmetic (e.g. source fitting, integration, histogramming)
6. Surface integrations inside curvilinear boundaries, for single and multiple images (different frequencies, polarizations)
7. Image slicing (profile creation) perpendicular to curvilinear tracks defined by user
8. Image filtering with standard (e.g. Sobel, unsharp mask) and user-definable filters
9. FFT and structure function analysis within user-defined sub image
10. Line channel combination and weighting to determine "continuum" image
11. Multi-beam spectral line cube imaging
12. Multiple line image "tiling" from cubes superimposed on, or plotted with lines to position in, source images
13. Multi-panel spectral line images

5.9.3 Three-Dimensional Tasks

Three-dimensional data sets are here described as "cubes" whether or not the co-ordinates are the conventional ones (two spatial + one frequency/velocity, two spatial + one temporal). The most important case is the spectral line "cube" since the time-variable cases are largely restricted to the solar or stellar domain. Cubes may be assembled from, or dissected into, images. AIPS++ must contains generalized facilities for such assembly and dissection.

1. Rendered surface displays of image cubes with rotation and aspect control
2. Interactive selection of position in an image with window display of local spectrum
3. Processing of 3-D cube of images of time variable sources using spectral line cube techniques
4. User-controlled image movie for single or averaged segments of line channels

5. Interactive slicing of cubes onto user-defined planes, with display and/or storage of output as image files

6. Interactive rotation of displayed cubes with "astrophysical" line of sight summation (e.g. radiative transfer)

7. Functional fitting in continuum cubes – rotation measure, spectral curvature

8. Extraction of spectra for selected regions with subsequent stacking, etc. operations as discussed under one-dimensional processing

9. Extraction and processing of frequency (velocity) profiles for specified spatial cuts in one- and two-dimensions

5.10 Self-Calibration Determination, Application, and Use for Editing

1. Phase or amplitude corrections to coherence function data sets based upon difference between the data and a model coherence function data set, where the latter is usually derived from imaging of from the same data

2. Editing on the basis of difference between data and self calibration iterations

3. Crossing-point array self-calibration (for some connected-element arrays, and for VLBI amplitude self-calibration)

5.11 Data Display and Recording

1. Flexible plots of Y vs. X (with optional dY and dX) where Y and X are any type of data in vector form, with point type, line type, and color differentiation for multiple X-Y plots

2. Contour plots of 2-D images with optional number labeling and width of contour sets, distinguishing negative contours and depressions, with color differentiation on appropriate hardware

3. Ruled surface plots of 2-D images, with possible color differentiation

4. Tiled displays of 1-d slice profiles or contour plots

5. Calibrated wedge displays

6. Pixel histogram displays in separate windows

7. Plot of pixel values in one image vs. pixel values in another, within user-defined boundaries

8. Image displays in windows with numeric and analog control of parameters, transfer function and color tables in other window(s)

9. User-definable color palettes

10. Display with histogram equalization

11. Intensity-hue display and independent RGB image superposition/comparison (on relevant hardware) of two or three images

12. Superposition of multiple coordinate grids on images – equatorial, galactic, super galactic, ecliptic

13. "Smart" superposition of contour displays on image displays (contours adjust grey scale or color depending on background)

14. 4-D display of images where intensity is a rendered surface and color on that surface is coded for a fourth parameter like polarization, spectral index, etc.

15. Snapshot hard copy of both separate windows and multi-window screen displays

16. Translation of image displays to input files for high quality color and gray scale copy devices, preserving transfer functions and color palettes where appropriate

17. Screen scratch pad capability including hard copy, allowing insertion of descriptive lines, boxes, shaded areas, and text on screen and hard copy; scratch pad applicability to specified frames of a movie

6 List of Tools

6.1 Existing Tool Packages

A number of existing software packages are known for their approach in providing "tools" that can be used to process data in many ways. AIPS, IRAF, UNIPOPS, Analyze, Gipsy, Miriad, Vista, SDE, etc. all have mixtures of tools and tasks in the sense discussed earlier. Relatively pure examples of tool packages are IDL, PV-Wave, AVS, apE, Khoros, etc. IDL and its offshoot, PV-Wave, are close to the ideal of a command language with a toolbox approach and a large degree of extensibility; they are best at dealing with processing and displays of 1-, 2-, and 3-dimensional arrays. They were not designed to deal with complex telescope data. However, we consider the tool capabilities and user interface for IDL/PV-Wave to be a slightly dated example of what AIPS++ should be in these areas. We believe that one should consider using other packages in an integrated manner with AIPS++ in order avoid duplicating many tens of man-years of software effort.

The distillation of tools out of a list of tasks is part of object-oriented analysis. The following is a user-oriented beginning at this distillation.

6.2 Data Base Tools

1. Single dish data base design

2. Visibility data base design with following elements:

   (a) antenna position information

   (b) information global to a "scan" (one continuous combination of hardware, antenna, and phase reference parameters)

   (c) Visibility data, weights, and quality flags for IF pairs and frequency channels for sequences at specific time and baselines (with u, v, and w)

   (d) Gain information with antenna-based intensity and polarization complex correction factors

   (e) History of processing steps that modify data

   (f) Instrumental data history during observing (monitor data for VLA and VLBA)

3. Catalog, file, and file system searching for data
4. Basic input and output for one or more visibility data base
5. Basic input and output for image "data base"
6. Multiple data base to single data base copying and/or averaging, with data selection
7. Formation of weighted and gridded u-v plane for selected data
8. Insertion of calibrator flux densities in u-v data base
9. Computation of simulated data for specified observing situation(s) and model source(s)

6.3 Data Import and Export Tools

1. FITS image reading and writing to/from image "data base"
2. u-v FITS data reading and writing with data selection
3. ASCII tabular file input and output with capability to skip reading selected number of initial lines
4. Sample data I/O programs that can be copied and modified so users can easily read or write different data formats

6.4 Data Correction, Calibration, and Editing

1. General data array calculus using constant/header data values
2. Correction of visibility data for predictable interferometer defects
3. Solution for antenna-IF calibration amplitude and phases to write in calibration table
4. Weighted interpolation/averaging of calibration table data and insertion in u-v data base gain information
5. Solution for polarization calibration parameters
6. Polarization position angle shift
7. Data flagging with both command line syntax and cursor interaction with data displays in t-Baseline-Amplitude or u-v-Amplitude form
8. Analysis and reporting of anomalies in visibility data for selected scans in terms of averages and their rms
9. Data flagging based upon deviations from visibilities derived from specified point source or intensity array models

6.5 Data Array and Image Calculus Tools

1. Association of images with names
2. Weighted linear image combination
3. Multiple image re-gridding and registration
4. Image source search and Gaussian fitting for dimensions and position
5. Primary beam correction in images
6. FFT transformation of real and complex (gridded u-v) images
7. Correction of images for u-v plane gridding effects
8. IQUV image arithmetic for polarization imaging
9. Image deconvolution with Högbom, Clark, and Cotton-Schwab CLEAN
10. Image deconvolution with MEM techniques
11. Image tessellation ("paste" images into larger scale images)
12. Image analysis of the fundamental assembly - the n-dimensional array or n-cube
    (a) May have only one pixel or may be 4K X 4K X 4K ...
    (b) Could be spectral line, time series cube (movie), with extra dimensions for polarization, ...
    (c) General tensor arithmetic in addition to normal mathematical operations
    (d) n-cube rotation, transposition, slice, stack, smoothing, block averaging
    (e) Extraction of n-cubes of lower dimension, including spectra for individual pixels
    (f) Identification of pixels or pixel regions to be treated as
        i. blanked pixel(s)
        ii. noise limited pixels
    (g) Fit m-dimensional surfaces with linear and non-linear techniques
    (h) Calculate divergence, curl, and Laplacian of vectors
    (i) On extracted vectors:
        i. fit baselines
        ii. fit Gaussians or other functional elements
        iii. cross-correlation
        iv. structure functions computation

6.6 Mathematical Processing Tools

1. General mathematical functions operable on number, vectors, and arrays
2. Least squares fitting, with error analysis, data vectors to ordinary and orthogonal polynomials
3. Spline fitting to data with associated use to produce interpolated data
4. Non-linear fitting of data to formulae using minimization of χ^2

7 Data Type Specifications

7.1 Data vs. Models

Data are fundamental in processing systems such as AIPS++. Previous sections have emphasized this in discussing data, tools, and tasks as fundamental elements. However, data are made sensible by the astronomer's model for what the data represent. One can mechanically describe the data produced by a telescope or telescope array, but to an astronomer the meaning of the data is dominated by the purposes for which they are to be used.

The next sub-sections will describe both data and operations on that data in general terms. This creates a problem that, in fact, permeates this document. The same words can mean different things for different instruments and under different circumstances. Ideally, we should be specifying everything in the form of mathematics. Mathematics is a language we should all have in common, so communication will be clearest, most specific, yet independent of implementation design and programming language. We have not had time to achieve the ideal of mathematical specifications. Some work is under way to do this, but in this document we must proceed with words rather than the ideal of mathematical models for data and data processing.

7.2 Data as Mathematical Operations on Radiation from the Celestial Sphere

Excluding the special case of space probes, astronomy is based upon observation of the distribution of radiation on the celestial sphere. The goal of astronomy is to measure and understand the set of Stokes parameters, IQUV, as a function of time, frequency, and position on the celestial sphere. All astronomical instruments are telescopes/detectors on some moving platform, whether the rotating earth, a satellite, or a free-flying space probe – and these measure some aspects of IQUV(t,ν,α,δ) or the equivalent mass and energy distributions of material particles; for present purpose we consider only observations of radiation. The theoretical problem for astronomical "objects" is then one of radiative transfer from and through those "objects" with physical processes that emit and absorb the radiation that does, or does not, reach the "celestial sphere". The astronomer's goal is to understand the nature, motions, rotations, etc. of the "objects" on the celestial sphere.

Astronomers build and operate telescopes and telescope arrays to collect information about IQUV(t,ν,α,δ). All telescopes collect radiation and eventually focus that radiation upon some detector. In optical and X-ray astronomy individual photons excite detector elements and are recorded by devices that reduce the information to n-pixel images, spectra, or combinations of the two. Radio telescope data sampling is best analyzed in terms of waves rather than photons, so one views wave fronts of IQUV(t,ν,α,δ) passing "through" the celestial sphere, and the earth's atmosphere, to be collected, focused, and sampled as electrical wave oscillations in some feed-receiver combination. The radiant energy is turned into fluctuating voltage(s) and is representable by an integral over IQUV(t,ν,α,δ) with weighting by an antenna response pattern $P_{\rm ant}(\nu,\alpha-\alpha_0,\delta-\delta_0)$ with implicit integration times and frequency sampling functions. Various amplifications and transformations occur before fluctuating voltages are: recorded as data, in the form of time sequences, spectral sequences, pointing sequences, etc. sampled as a function of time in a computer; or are cross-correlated with comparable data from other telescopes before sampling and recording. Data processing in the context of AIPS++ begins with the resulting instrumental data sets, which represent complex transformations of the information the astronomer is interested in: IQUV(t,ν,α,δ). Secondarily there are other types of data that can be important in processing instrumental data sets. The first are instrumental parameter data - pointing positions, focus information, sprectrometer response to "white noise", equipment parameters, etc. These secondary, but very important, data are diagnostic information for instrumental components that can be used to evaluate data quality or, in some cases, make corrections to the data (e.g. system temperature). All of these data may need to be used by the astronomer studying particular phenomena in objects seen on the celestial sphere. As discussed earlier, data need to be analyzed both during and after the observing process - this has always been true for single dish observations, and we must plan for this for current and future arrays.

7.3 Total Power Data Sets

The basic data from a single telescope or phased array of telescopes are time, spectral, and/or pointing series of total power measurements for particular polarizations, frequency channels, and integration times associated with telescope/feed/receiver states. One can use a notation P[t,p(t),ν(t),α(t),δ(t); s(t), $\Delta\nu$, Δt] to identify these data with their basic parameters and an instrumental "state function" $s(t)$ (focus and other parameters affecting the data). The latter represents the need to associate the data with differentiable states based upon beam, position, and/or frequency switching. The instrumental state function and the total power data can then be processed to obtain, edit, and calibrate meaningful measurements of sky intensity, in the form of antenna temperatures. Thus there are four classes of data: the total power data sequences; the instrumental "state function data"; instrumental parameter data; and instrumental diagnostic data that need to be handled for the astronomer or achieve his purposes.

The principal types of single dish data are:

1. One-dimensional spectra – both evenly and non-evenly spaced (e.g., taken with AOS spectrometers) so a second one-dimensional array identifies spectral frequencies

2. One-dimensional continuum scans – usually unevenly spaced and associated with an array of pointing positions

3. One-dimensional arrays of data values taken at arbitrary positions, times, foci, etc. – such data type are for tipping, continuum on-off, focusing, "five-point" pointing, etc. observations (1-D spectra and scans are a subset of this type of data)

4. Two-dimensional matrices of data values as a function of (x-position, y-position), (position, frequency), (frequency, time), (position, time), (time, pulsar phase), (pulsar phase, frequency), etc. where both axes may be non-linear or non-parameterizable

5. Three-dimensional cube of data values as a function of (x-position, y-position, frequency), (x-position, y-position, radial velocity), (time, frequency, pulsar phase), (x-position, y-position, time).

Sometimes intensity is not the data one is analyzing. For example, analysis such as baseline fitting, smoothing, and plotting 1-D arrays with the position of the telescope (the 'data' or dependent variable) as a function of time (the independent variable).

Superimposed upon the instrumental and total power data are the oft-debated "scan" and "sub-scan" concepts which has a meaning imposed only by the astronomer and his intended use of the data. In some cases the meaning relates to instrumental corrections, such as pointing sequences on strong sources with known characteristics. In some cases the meaning relates to calibration of instrumental parameters that vary as a function of time. In all cases, the astronomers understanding of how the data is to be used is part of an editing process where good or acceptable data are differentiated from bad or unacceptable data.

7.4 User-Oriented Data Organization

From the point of view of the user the highest level identification of the problem is what we will call a "project". Projects are aimed at obtaining answers to scientific questions. Answers to these scientific questions frequently involve obtaining data from a variety of telescopes. Some projects require radio data from both single dish and array observations from the same or different instruments, each serving a different "purpose". Observations for each instrument are organized into observing "runs" with sequences of "scans" as discussed in the last sub-section. Each scan contains "sub-scans" with data elements in the form of spectra, time instances of coherence function data or spectra, etc. that are associated with instances of time. Astronomers need to deal with this hierarchy of data: project, purpose, instrument, observing run, scans, and sub-scans. It would be very helpful if the astronomer could be aided in dealing with things according to this hierarchy. Data that are viewed as simple sequences of data from stand-alone telescopes leave the astronomer to impose a mental image of project/instrument/purposes and then runs/scans on the simple data elements. The future mmA will be a case where the same instrument will generate both single dish and coherence function data sets. This makes it a prime example where the same instrument will serve diverse instrumental purposes for a wide variety of "projects".

In an earlier section listing tasks, preparation for observing was mentioned. This is partly because simulation, using AIPS++ processing tools, can be very useful in understanding an observing program during the planning and preparation process. In addition, it is at this stage that the user imposes the logic of project/instrument/purposes/runs/scans on the observing process, and this logic must be remembered and used as part of the data reduction and processing. If tools were available in AIPS++ to aid the user in passing on and using this logic all the way through data processing, it would be very helpful. It would be analogous to having and updating the map of a maze that can be used while passing through the maze. Data processing is very much like a maze to be negotiated for most astronomers, and assistance in dealing with the higher level purposes of data would be very useful.

7.5 Coherence Function Data Sets

As discussed in the previous sections, coherence function data sets often cannot be discussed independent of data from other instruments. However, they have special characteristics and complications – not the least of which is the large volume of the data sets.

Arrays of radio telescopes, with either real-time phasing or data recording with appropriate time identification, are cross-correlated after appropriate time delays to record coherence function data sets. For N telescopes with sampled polarizations p this leads to up to $N(N-1)/2$ averaged (in time) samples of visibility (coherence) functions for particular times, frequency pass bands, and sampling coordinates determined by the projection of antenna-pair separations on the celestial sphere at the point where the antennas are pointed. Coherently phased arrays like the VLA are corrected for delays, and phase adjusted, before correlation with phasing with respect to a particular position in the antenna beam. Data recording arrays, using VLB techniques, produce data tapes that are later correlated with equivalent assumptions about antenna-pair separation and phase reference position. A fundamental assumption is then made that adjacent IQUV(t,ν,α,δ) on the celestial sphere are *not correlated with each other, so the product of their time-averaged correlation is zero.* In that case the instrumental coherence function data set is a two-dimensional Fourier transform of the celestial sphere IQUV(t,ν,α,δ) weighted by antenna response patterns.

The so-called interferometer equation describes the two dimensional Fourier transform of IQUV(t,ν,α,δ) on the celestial sphere, with weighting by antenna beam shapes, that results in visibilities with instrumental defects. These visibility data, instrumental calibration information, instrumental performance data, data editing information, and the instrumental parameters for "scans" constitute the basic data for connected-element interferometers that must dealt with. However, they can be based upon assumptions that are invalid for tape-recording arrays (VLBI), as we will discuss in the next section.

7.6 VLBI Coherence Function Data and Imaging

Coherence function data from VLBI correlators are derived from cross-correlation of data with delays that are never known to sufficient accuracy to allow "simple" processing. For this reason correlation is done for a range of delays or time lags, and the results are transformed to the frequency domain so VLBI correlator data are in the form of visibilities for a range of frequencies. Different techniques are then needed to fit to residual fringe rate errors and spectral slope. "Global" fringe fittings are done for an entire source data set and used to solve for $d\phi/dt$ and $d\phi/d\nu$, where ϕ is the visibility phase for an antenna pair. VLBI data from modern correlators are always spectral line, whether there are spectral features or not. In addition, improvement of astrometric parameters for telescope locations and time corrections can require re-fitting of some data at much later times. This application therefore requires data archiving in a special data base format, and imposes strict requirement for recording (with the basic data) detailed information about telescope parameters and all corrections that have been applied.

The long baselines of VLBI cause the field of view that is not radially smeared by the range of frequencies in each frequency channel to be very small, typically less than an arc second. In cases where sources contributing to the visibilities are spread over more than a single field of view, such as for masers, "fringe rate mapping" is a necessary technique whereby individual sources are imaged at different fringe rates - which is equivalent to changing phase reference position for each pixel in an image. One way of looking at this process is doing a direct transform on the data to get a single pixel intensity with phase reference with respect to the location of each pixel, or in practice, zone of pixels inside a "field of view".

Although there are no fundamental differences in the conceptual model of a VLB interferometer as compared to a connected-element array, VLBI has more stringent requirements. The baselines are extremely long, the coherent integration times are short, the (bandwidth limited) fields of view are very small, and it is very sensitivity- and dynamic range-limited. The data sets are large because of the requirement of short integration times before fringe fitting, and because the VLBA (and all modern) correlators produce large numbers of spectral channels for each of these short integration times. Greater accuracy in equation approximations and more complicated models are required because of the large antenna separations and needed time and geometry standards. VLBI data processing in AIPS has been limited by the degree to which assumptions valid for the VLA, but causing problems for VLBI, were originally built into the system.

In the case of VLBI with antennas in space, the parameters of the problem become factors of 10–20 more extreme. Certain approximations for time and Earth geometry cannot be used, even though they are valid for a locally confined array of antennas.

For these reasons, the standards of accuracy for interferometric array data processing should be set by the VLBI requirements, since the needs for other arrays are less difficult to achieve.

7.7 Imaging and Subtraction in Coherence Function Domain

Beyond the traditional imaging computation, MX in AIPS is a major example where combinations of imaging, subtraction of models in the u-v plane, and image deconvolution were very effective. Flexible u-v data subtraction is important to many frontier problems. Wide field, 90cm spectroscopy with the VLA has depended on mating new techniques of u-v subtraction and standard image computation. Image calculus is a concept that is well developed because it is amenable to well understood mathematical operations on vectors and arrays. However, visibility data calculus is still in a rudimentary state. One example that has recently gotten special development is imaging with a combination of total power and coherence function data: the mosaicing problem. Fourier transforming total power images, and dealing with images and visibility data from both domains, are areas where tools are needed to expedite both algorithm development and efficient scientific use in this area.

7.8 Mosaic Data Sets and Mosaic Imaging

An important example of organizing coherence function data sets according to a particular purpose is the case of mosaic imaging where total power and coherence function data are taken with systematic sampling of a number of antenna pointing centers. Individual data points must then be labelled in some way with a specification of a pointing center, just as they are labelled with u,v,w coordinates currently.

Currently, many mosaic capabilities exist within AIPS and SDE on a "pointing-by-pointing" basis; the visibility file for each pointing can be self calibrated, edited, etc. The pointing-by-pointing capability must be kept for AIPS++, but an "all-pointings-at-once" approach is needed for most operations. Consider, for example, the task of editing and calibrating each pointing of a 1000 pointing mosaic. In addition, it must be very easy to specify a subset of all pointings for a given operation.

Although a mosaic dataset is somewhat different from that in an ordinary visibility or total power dataset, the final mosaic image will be either a two dimensional image or a spectral line data cube and has no special status.

The development of mosaicing is still in its infancy and it is likely to progress further both in understanding and in techniques once a better means of expressing the algorithms becomes available. Hence this is an area in which we envisage a substantial continuing evolution of requirements.

7.9 Multi-Spectral Window Data

Current single dishes, and millimeter arrays such as BIMA, routinely observe with simultaneous sampling in different spectral windows, and these windows are often wide enough so that a large number of spectral lines are present. This leads to special image cube analysis problems, and in the case of arrays the possibility of special self-calibration techniques. For example, if there is a strong line in one portion of a spectral window, one can "self-calibrate" to remove the atmospheric phase variations using data in that window. This technique will allow large increases in coherence or integration time.

7.10 Image Deconvolution and Analysis

The images computed from Fourier transform of coherence function data are "dirty" images because the sidelobes generated due to missing visibility data "smear" the source in the images.

Without "correction" for the effects of these sidelobes, images would be limited to the order of 30-50 to 1 dynamic range. The Högbom, Clark, and Cotton-Schwab CLEAN algorithms amount to a least-squares fitting in the image plane of the dirty image to a model of N point source functions, with the CLEAN image resulting from a sequence of subtractions of these point sources in the image plane in the form of the dirty beam; when the subtraction process is complete, the CLEAN image is the resulting residual map with the point sources restored with a 2-D Gaussian beam (or some other function) with the same shape as the core of the dirty beam. This process is currently the most common form of image deconvolution. Other fitting procedures can be used to generate "models" of the sources in the image. Maximum entropy deconvolution is particularly advantageous for sources with broad emission structures, while CLEAN is advantageous when the image is well-represented by point source models. As mentioned earlier in this document, algorithms are the things most like to change in the future, particularly if AIPS++ software is designed for good algorithm development. Eventually even hypothetical algorithms like Singular Value Decomposition may be developed, representing the images as a linear superposition of base-set "images" - a process that has not yet been used in radio astronomy.

In the previous discussions all images were described as 2- or 3-dimensions in space. Actually, an n-cube representations for IQUV with extra dimensions involving frequency, and possibly time, satisfy the same mathematics. A special case is n-cubes where one of the dimensions is time and there are sources in the image varying significantly during the time the visibility data are obtained – a circumstance common for solar and stellar radio sources. In this case imaging and image deconvolution must be made with short time intervals over which the IQUV(ν,α,δ) can be assumed to be constant, and sequences of dirty or deconvolved images as a function of time constitute the time-dimension of the n-cube.

As discussed in the sections on tasks and tools, the extraction of information from n-cubes is one of the major tasks for AIPS++ once one begins working in the image domain – a process called image analysis.

8 Image Analysis

8.1 Traditional Image Analysis

Traditional image analysis involves the extraction of astrophysical information from data sets of two or three dimensions. Thus the image is an n-cube as described above (actually an n-rombohedron might be more accurate). In each plane the intensity on the sky is encoded - usually the actual brightness, but sometimes polarization angle, velocity, etc. Usually the data meet the naive definition of an image - a picture of an astronomical object - so the first two dimensions are spatial. The third dimension is usually velocity, frequency, or time. Traditional images will continue to be the primary material for image analysis, but there is room to advance the science by analyzing other sorts of images in which one or both of the first two axes are not spatial or the data set has more than 3 dimensions. Question that should be addressed are: how much quantitative analysis of l-v diagrams can be done; are there problems best solved by considering polarization as a function of position and frequency; etc. VLB imaging in which one dimension is defined by frequency gradient represents one area where n-cube analysis and fitting could be used.

The most traditional function which will remain essential in AIPS++ is astrometric analysis: determination of position angles, locations of intensity minima and maxima, etc. This will be easy or difficult according to the way coordinates are carried with the images. Compared with other common astronomical/scientific packages this is one of the strengths of the current AIPS. Additional coordinate systems such as radius (or the log of the radius) and azimuth in the plane of a galaxy would be useful. This is currently done using LGEOM or PGEOM.

Related tasks are those which require position centers or boxes for input-aperture photometry and aperture-based editing being obvious examples. Boxes, or more generally, regions, could interchangeably be specified by coordinates, by pixel numbers, or by movement of a cursor on a TV display. The total signal within the region, and pixel statistics, should be returned. Background subtraction from surface fitting or additional regions should be possible. This brings up the problem of blank pixels, zero pixels, background noise determination, and analysis of the significance of pixels near the mean noise level. In many situations the option of blanking is helpful; moment analysis of spectral line cubes is an example. This is preferable to setting the pixels to zero if the image already contains physically meaningful pixels with zero or near zero intensity - the blanked pixels have an identity separate from that of the zero pixels. The most flexibility in blanking/zeroing would come from increasing the dimension of the n-cube or creating a second cube and including display and analysis routines which allow options for comparing the original and blanked/zeroed regions.

To assist in blanking/zeroing statistics, packages which could operate on regions would be essential. Along with the familiar mean, median, mode, standard deviation, and rms should come more recent developments allowing survival analysis and noise modeling for χ^2. Nonparametric statistics might be helpful here. Gaussian and boxcar smoothing would also be required.

Additional help could come from a flexible histogram routine for deciding how to edit and display data. Occasionally pixels, lines, or regions would have to be edited or values replaced. This should be flexible enough to include interpolation in one or several dimensions. Noise with a spectrum representative of several types of instruments could optionally be included to make the interpolations less obvious visually. Filters, such as max/min, mean, or median should be available.

For combination of images several options should be handy - simple block tessellation, stack filtering (min, max, sigma clipping, median, mode). Image registration is a must, by rotation, transposition, reflection, and x-y shifting. Exceedingly demanding projects would be helped by a convenient routine for generating test images - images with all pixels of intensity one, of known FT, etc.

Some n-dimensional data sets have more complicated parameterizations of one or more axes. For example, continuum data taken with a single dish as the telescope starts to move, comes up to speed, if blown "off course" by a gust of wind, and slows to a stop. Some cases are more predictable, such as gaps in time-sequence data during "data invalid" periods when equipment is changing, antennas are not yet on source, etc.

8.2 Spectral Image Analysis

Earlier sections of this document have presented the idea that spectral analysis is often independent of the data source. Thus determination of line centers, baseline subtraction, Gaussian fitting, de-blending, and moment analysis should all be available. The optical community has made great advances in the analysis of two-dimensional spectra (from long slit and multi slit CCD instruments). This should be part of n-dimensional spectroscopy in AIPS++. More and more VLA spectral line projects include both emission and absorption data, so it should be possible to easily combine this information.

In spectroscopy more than any other discipline modeling is central to an understanding of the astrophysics. Built-in modeling routines, particularly radiative transfer, would be very helpful in this area.

8.3 Non-traditional Image Analysis

AIPS++ should provide the opportunity to advance the science. Rather than trying to guess what the next efforts will be, every effort should be made to provide easy access to as many standard mathematical techniques as possible to allow astronomers to hone their own cutting edge applications. Some projects currently under way in the AOC are examples: rotation and registration of solar images taken at different times and locations on the Earth is necessary, but not easy, because the Sun is not a rigid rotator. Treating VLA HI images of galaxies as snapshots of fluids means the divergence, curl, and Laplacian must be calculated to study continuity, vorticity, and viscosity.

9 Image Display and Recording

Image display and recording serve two purposes closely linked to visualization: they allow communication of the scientific content of an image (and the results of its analysis) and allow inspection of the data calibration/reduction process for quality control and scientific inspiration. These are essential operations - there must be tasks which allow them. This is a case where tools alone will not do the job - tasks like TVFLG in AIPS would be difficult for the scientist to build with primitives. In addition, there is a need for tasks which display the u-v data directly for real-time or automated inspection and flagging.

9.1 Image Display

Currently the Image Storage Unit (ISU) is the most powerful image display generally available at NRAO. This hardware is a very powerful, flexible display system. The tasks for color and black-and-white contrasting, nonlinear transfer functions, intensity-hue, pseudo color, overlays, movies, split screens, and histogram equalization are excellent. Despite this the simple TVALL, TVMOVIE, TVPSEUDO, etc. in AIPS get much more extensive use because they are much easier to learn and remember. This is a user interface problem - if learning the ISU and storing images were easier more would use the ISU. Some use the ISU extensively and supplement it with GIPSY tasks such as NINER, indicating that GIPSY still has several display/analysis tasks which out perform those in current AIPS and the ISU by such a wide margin that it is worth the effort to learn to use an additional package.

The capabilities of the ISU and GIPSY should be available on workstations; test versions of software on loan show this is available now. We should be able to rotate cubes (the Sun utilities have an impressive task which displays cubes using pseudo optical depth, and the cubes can be rotated), take slices or tilted planes through cubes, differentiate, choose complex transfer functions, have edge finders, unsharp masks, etc.

Along with the display should come tools for image statistics and line graphics, 1-D image cuts, and histograms to help determine the optimal parameters for display. GIPSY and IRAF have some of these tools built in.

9.2 Image Recording

Currently images are recorded on film using the Dicomed at the AOC and a newer camera system in Charlottesville. The Dicomed gets the job done but is slow and requires a lot of staff and astronomer attention. Several currently available devices are much simpler for the scientist and require less maintenance. Images are also stored on video tape using the dedicated system on one of the AOC Convex machines, but despite the ease of this process and the high quality of the results, this capability is rarely used. Video is the best way to present a spectral line cube or a time-dependent model in a colloquium. The current capability is probably adequate given the tools available and the demand for them.

Advances in image recording could greatly improve the diagnosis of image problems and support of visitors. Imagine a history utility which records all input files and allows re-execution (and de-execution to undo steps). The advanced user could replay and redo steps to determine the source of a processing error. The addition of a tool to make snapshots of screens and images would prove most useful for visitor support. A visitor could order the snapshot saved in a file which could then be displayed at any workstation - even over the Internet - so an image processing guru, local or not, could be consulted. The expert would simply open an additional window and would not have to suspend all personal work to give advice.

10 User Interface

Some aspects of user interface were discussed above as part of the general attributes of AIPS++. In this section we emphasize an important subset of these attributes.

AIPS++ should have good command line interface with "full" programming capability. This should be at the level to eliminate, for most astronomers, the need to write FORTRAN or C++ programs. We view the issue of who will be able to develop applications programs in AIPS++ as one of the most important issues for the future. "Full programming" capabilities with the AIPS++ "command language" is very important; however, at some level the use of C++ and FORTRAN "template" programs that can be run "with" AIPS++ is also important. In addition, the current plan to have many astronomers doing C++/OOP programming for AIPS++ will require special attention to astronomer-oriented documentation, programming guides, and things like programming "summer schools". Assuming that everyone can learn from industry-wide material for professional programmers is unwise, and is likely to limit the AIPS++ pool of developers to too small a group with too little astronomical experience.

AIPS++ should be operable from purely ASCII terminal or terminals/emulators with ASCII and "Tektronix" emulators; however, this should not be a reason for limiting the use of graphical user interfaces (GUI). A low level of GUI should be the primary interface for the majority of users. The lowest level of GUI should be X-windows compatible allowing multi-window command line input, menus, help information, plots, image displays, etc. – this would accommodate PC's with X-windows emulators and, most importantly, the modern workstations that are becoming nearly universally available. It would be very desirable if there were an AVS-like GUI for accessing, querying, specifying parameters, and chaining of processes. The command language should allow pipelining of applications and standard data flow between "tools" or "tasks". One also needs "batch" command line programming with monitoring, interrupting, and redirection capabilities.

Parameters for applications should be named with conventions familiar to astronomers, with flexible means of specification by command line, editing of "inputs files", selection from menus, etc. Global parameters or variables should be minimized. Some global parameters specifying packages to be use, input and output environment, etc. are necessary, but we recommend that scope of parameters for applications be localized to specific applications.

Documentation should be available both on-line and in hard copy. This should have multiple levels ranging from simple "help" to extensive information. Consistency between hard copy and on-line documentation is imperative. Multi-window environments, as mentioned above, should allow context-sensitive information to be displayed by "clicking" on appropriate items. While the implementation aspects of a UNIX "man" page might be useful, the displayed information should be easily understandable to user-astronomers.

Multiple levels of user interface would be desirable to allow for both novice users and experienced expert. User selection of the style of interaction and the range of "packages" to be used should be possible.

Styles of user interface are difficult to decide upon, and are very dependent upon user experience and preference. The discussion in Wood (1991) is an example of a useful approach to the user interface that goes into details we have not discussed here. We recommend planning a number of available styles, and extensive user testing of each of them during early phases of AIPS++ development, as opposed to deciding upon one approach and precluding all others. The idea that the user interface is just another applications task, that can take many forms, is probably very important in planning for the future with a wide range of user needs and expertise.

11 Priorities

11.1 General

Both in the design of classes and in the initial implementation we recommend that first priority be development and optimization for spectral line work for both single dishes and arrays. The temptation to initially emphasize continuum data processing, and then implement spectral line as a number of narrow continuum channels, should be strongly resisted; historically it has resulted in cumbersome, slow processing of spectral line data for arrays. Single dish data analysis should be amongst the first application areas to be made operational; this is partly to insure support for this area, and partly because the appropriate model for spectral line spectroscopy for arrays should be the type of interactivity with the data that has been traditional for single dish operation. We strongly recommend that data simulation programs be part of initial development in these areas so software can be tested under "full load" conditions and with predictable results.

11.2 NRAO-Oriented Single Dish Priorities

The time scale for AIPS++ applications should match the 1995 date for the initial operation of the Green Bank Telescope. The development of new single-dish software is an urgent need for the GBT, but the needs of the 12m and 42m telescopes should also be considered. Indeed, the latter NRAO telescopes can be viewed as test-beds for AIPS++ software for not only the GBT, but also the single dish data processing for the future mmA. However, the GB/CV based single dish development for AIPS++ will be preeminently software for a new instrument – the GBT.

11.3 NRAO-Oriented Array Priorities

By at least a two to one ratio, it is the view of the astronomers at the AOC who have been participating in the AIPS++ Development Seminar since August 1991 that the highest priority for applications development at this site should be in the area of calibration, editing, and related data evaluation. For the VLA the goal will be to reduce the limitations of the current software for these purposes. For the VLBA it is because there are fundamental data integrity and data processing problems that will demand solutions as the VLBA evolves from testing to routine operation. The improved programmability of AIPS++ should make it the development environment of choice in 1993 and beyond.

As mentioned earlier, the accuracy of approximations and the data base standards for dealing with array data should be set by the requirements of VLBI data processing. From NRAO's point of view, AIPS++ is being developed in the middle of the completion and initial operation of its newest instrument, the VLBA. With VLBI, dealing with properties of the instrument is inextricable from correlator post-processing. The long term success of the VLBA as a user-friendly instrument for non-VLBI specialists will critically depend upon the quality of the software provided for VLBA/VLBI in AIPS++. We therefore believe it should be an urgent NRAO priority to develop VLBA/VLBI processing capabilities as early as possible in the development of the AIPS++ project. In addition, since the VLBA will operate with other telescopes a large fraction of the time, it is important to extend the scope of VLBA software to include other elements, including orbiting elements.

12 Further Development of Specifications

The discussions at AOC seminars and the iterations on drafts of this document make it clear that based upon background, interests, and philosophical preferences, there are a wide variety of views towards written specifications for software. This document is just one beginning document amongst those that will be considered during the design phase of AIPS++ in Charlottesville in January-June 1992. Additional views on specifications are welcomed, particularly well-posed ones appropriate for the Specifications Memo series.

Amongst the major problems that need to be solved for the AIPS++ project is use of a common language. The same words have different meaning for single dish telescopes, connected-element arrays, and VLBI arrays. Resolving these terminology problems will be essential for a software system used for many different telescopes with diverse purposes.

The process of communication between AIPS++ programmers and astronomer-users must be continuous. In some sense these specifications are simply a beginning of a process that will continue during the Charlottesville design phase, and indeed beyond when user feedback becomes important to "de-bugging" and improving the software. It is clear that there will always be a communication problem to be solved because of the need for involvement of experienced astronomers in the evolution of specifications, the dialogue to evaluate the results of object oriented analysis, the development of applications, and the testing and use of AIPS++. Given that most experienced astronomers will find it difficult to learn to work in the C++/OOP environment, we recommend two positive actions to best utilize the experience of astronomers. The first is that one member of the "full-time" AIPS++ group at NRAO function as a project scientist whose primary responsibility will be paying attention to, and analyzing feedback from, user astronomers. The second is that, when applications development begins, some AIPS++ programmers be assigned to work closely with astronomers who are experienced in software and algorithms but do not want to spend most of their time writing software; this will in part bridge the generation gap between the old and new ways, and in part will be an effective use of both astronomical and programming expertise.

13 References

Cornwell, T.J. 1990, *Report of the Software Advisory Group*, AIPS++ User Specifications Memo 102.

Foster, R., Haynes, M., Heyer, M., Jewell, P., Maddalena, R.J., Matthews, H. Reich, W., and Salter, C. 1991 *Requirements for Data Analysis Software for the Green Bank Telescope*, GBT Memo 72.

Wood, D.O.S. 1991, *The AIPS++ User Interface*, AIPS++ User Specifications Memo 104.