N93-24553

Systems Aspects of COBE Science Data Compression

I. Freedman, E. Boggess, E. Seiler (Hughes STX).
Cosmology Data Analysis Center
Greenbelt, MD 20771

Abstract. A general approach to compression of diverse data from large scientific projects has been developed and this paper addresses the appropriate system and scientific constraints together with the algorithm development and test strategy. This framework has been implemented for the COsmic Background Explorer spacecraft (COBE) by retrofitting the existing VAX-based data management system with high-performance compression software permitting random access to the data.

Algorithms which incorporate scientific knowledge and consume relatively few system resources are preferred over *ad hoc* methods. COBE exceeded its planned storage by a large and growing factor and the retrieval of data significantly affects the processing, delaying the availability of data for scientific usage and software test. Embedded compression software is planned to make the project tractable by reducing the data storage volume to an acceptable level during normal processing.

1. Introduction

Large scientific projects generate diverse scientific, engineering and instrument housekeeping data at rates that frequently exceed the capacity of storage and retrieval devices. Although many techniques have been proposed in the data compression literature [1], almost all are based on data models that make predictions based on a few successive pixels or a few hundred images in a training set. These data models do not incorporate the *a-priori* scientific knowledge of approximate relations between data set elements (physical laws) or the known accuracy requirements for specific elements of record structures. Such knowledge reduces the specific entropy of the data, enabling an effective trade-off in wall-clock processing time between additional cycles for on-the fly compression and decompression and a reduced input-output load.

If the system response is sensitive to the network load (when the network is saturated) reduction in storage complexity may be as critical as reduction of the overall load. Furthermore, fixed mechanical disks are an expensive resource and the risk of catastrophic data loss increases dramatically with the number of disks on the system. Local SCSI disks are sometimes suggested to represent inexpensive storage media but the access time is relatively long. Mass storage devices such as magnetic tape juke boxes can be less than ideal as the tape quickly stretches with use and becomes unreadable after a short time (1 year) compared to the typical project lifetime (20 years) necessitating frequent and expensive data migration.

2. COBE Science Goals and Achievements.

The COsmic Background Explorer (COBE), NASA's first satellite devoted to the study of cosmology was launched on 18 November 1989. The cryogein period of the mission covered the time from 21 November 1989 to 21 September 1990. COBE carries three instruments: the infrared experiment DIRBE, the anisotropy experiment DMR and the spectrum experiment FIRAS, of which DIRBE and DMR are still operating [2].

All three instruments have achieved their preliminary goals. FIRAS has shown the far infrared background to be isotropic to 0.03% and consistent with a black-body radiating at 2.726 K.[3] DMR has revealed further evidence of the Big Bang theory of cosmology in the form of a spectrum of ripples at the level of 1 part per million after known astrophysical foreground sources have been subtracted from the integrated signal. DIRBE has placed upper limits on the spectrum of the diffuse celestial background which are more stringent than previously available [4]. DIRBE and FIRAS have contributed to Galactic astronomy by mapping the stars in the direction of the Galactic core [5], modelling the physical conditions in the interstellar medium [6] and making a determination of the radial distribution of NII ions [7]. DIRBE has also contributed to interplanetary astronomy by providing an accurate phenomenological model of the Zodi Light from the interplanetary dust cloud [8].

Figure 1 shows the DIRBE annual average 100 micrometre map which is an example of the most detailed map data with highest contrast and largest dynamic range.

3. Ground Segment Architecture

The ground segment computer architecture consists of a VAXcluster linked by an Ethernet network bridged by a hardware-based repeater. It supports approximately 100 users in the daytime, production work at all hours, and system management and monitoring activities [9]. The HSC's serve 100 Gbytes of magnetic disk to the cluster, which consists of four mainframes and thirteen workstations. Interactive development and analysis work is done on the workstations which provide almost all the CPU power in the cluster. The mainframes are reserved for disk serving and batch processing. With the advent of truly high performance workstations, the I/O demands are also increased and disk serving has become a critical load to all but the most powerful mainframes. Two DECStation 5000 workstations are currently available and are linked to the VAXcluster using NFS. The data sets generated by the project *pipelines* are available to remote users and PCs through a *data server* and can be manipulated using IDL which is in widespread use on the VAX/VMS platforms.

4. Project Data Sets

The COBE satellite carries three experiments designed to make high precision measurements of the diffuse celestial background.

The detectors are stable and data sampling highly redundant. The observed sky is faint, low-contrast and smoothly variable except for one instrument (DIRBE) which sees stars at fixed map coordinates. FIRAS and DIRBE report glitches, many of which arise during passages over the South Atlantic Anomaly region.

The processed data currently totals 380 GB with an effective expansion factor of (4-16) over the raw data which depends on the instrument system. The project standard data sets number about 1000 and may be classed as sky maps, time-ordered data and time-tagged data. These data sets combine scientific with engineering data.

The Project Data Sets are required to represent data free of instrumental signature and the Analyzed Science Data Sets are intermediate to further scientific interpretation.The Astronomical Databases [10] contain external survey data converted to the COBE sky cube pixelization scheme [11],[12] at the resolution and beam pattern of the COBE instruments. The sky cube is an approximate equal-area projection on the sky of the faces of an inscribed cube. The equal-area property is ensured by the curvilinear coordinate system ruled on each cube face.

COBE data sets are directories of files. Intensity, spectral and polarimetric data are stored in area quadtree maps together with ancillary information. Offsets into each map corresponding to each level of resolution are stored in "index" files. Each pixel may contain one or more records with the same field structure.
Data destined for the DIRBE experiment are stored at sky cube pixel level with 9 or more levels of resolution available in an image pyramid obtained by spatial averaging; data intended for FIRAS and DMR are stored at 6 or more levels of resolution. Data records are fixed length, defined by a Record Definition Language (RDL) file interface to the VAX Common Data Dictionary. RDL and its Record Definition Compiler were developed by the COBE project [9].

Figure 2 shows an example RDL for the DIRBE Daily File.

5. Data Compression Requirements

Data Compression is intended to simplify the task of systems management, data migration and recovery from catastrophic disk failures, reduce expenditure on storage devices and improve the data retrieval rate by a substantial factor dependent on the non-linear response of the saturated network.

The COBE Ground Segment Software System [9] consists of approximately 500 packages known as *facilities* which process the data in *pipelines* for each subsystem from raw telemetry to Project Data Sets. Access to the data is provided by the Data Management subsystem heavily dependent on a project-specific access system known as *COBEtrieve*.

Interviews with the Principal Investigators and Contract Leaders defined requirements as follows:

Provide compression transparently without changing the application software.

Compress instrument pipeline and science analysis data products to better than (16 to 50)%.

Process compressed data at a throughput not less than 90% of uncompressed data processing (possibly faster).

Preserve required accuracy of instrument housekeeping and scientific data (as judged by validators).

Exceed bitwise reliability of 10^{-13} on average (flawless compression of 300 GB). Several times this factor is desirable.

Support full random access to file records.

Provide a capability to select specific classes of data for compression.

Preserve overlaps in separately-processed data segments.

Store search keys (time code, pixel address) in clear codes.

Provide a capability to select a compression scheme for each field of a data record.

Optimize choice of compression scheme combining *a-priori* with adaptive knowledge of data.

6. Implementation

Initial tests with public domain software (Unix-compress) and commercial PC-based hardware (Stacker, a product of Stac Electronics) demonstrated poor performance. The software was far too slow to keep up with the processing and Stacker compressed the DIRBE Daily Files (the largest archived files with the greatest retention time) by < 2%. Although the offlining of disk volumes provided by the FlashDAT 4mm tape device (a product of Winchester Technologies, Inc.) has been highly effective (factor of 4 improvement in data migration rate with a compression factor of 2), the requirements listed above necessitate customized software.

The following decisions were taken :

Create standalone, callable and embedded software interfaces.

Use existing fixed-length file record structure.

Use existing search algorithms to retrieve data.

Store compression parameters in file header without increasing the number of open files. This averts a resource lock-management problem in an already full file system.

Adopt an incremental build strategy: simple, well-trusted algorithms followed by powerful sophisticated methods.

Assess all algorithms on samples of all types of project data : [Quadtree Sky Map, Time-Ordered, Time-Tagged].

Optimize tradeoff between throughput and compression factor via overall measured storage savings.

Store data for medium-term via deeply-compressive but slow methods assisted by accelerator board (single files recovered in << 8 hours).

Offline project data via hardware-based compression methods.

Data shall not be delivered in compressed form to external users.

7. Compression System Design

Since the data access is heavily dependent on *COBEtrieve* and all the I/O system calls are localized, a natural solution is to embed compression software between the data management and I/O layers. This software compresses and decompresses data from the stored format to the fixed-length record structure understood by the data management software.

The writing of compressed data may be toggled via a system-wide logical name. The reading of compressed data is always enabled. The compression method specification is via command-line qualifiers which may be stored in the compressed file header and parsed to control the decompression of archive files. These qualifiers drive the command line, callable and embedded interfaces uniformly.

Currently, the record length and connect-time attributes of recognized standard data sets are stored in a VAX Datatrieve data base (DAFS). When a file is opened, this data base is queried and if the data set name and record length are matched, the data are accessed. Separately-processed time-overlapping data segments are stored in separate files but the data streams are merged based on the most recently-processed data from each segment.

Similarly, we may define a compression data base (CMPR) that specifies the command-line qualifiers (including the record length) which will be parsed to control the (de)compression of archive files. Since multiple compression method types and offset endpoints are defined for multiple offset ranges, this data may require updating on every change of data set Record Definition Language specification. Ideally, this information would have been provided by the scientist when the data sets were being designed.

The compression system permits full upwards and downwards compatibility with existing files and catalogs. If the record-length matches the entry in the DAFS data base, file is assumed uncompressed. If the entry does not match, the file header is parsed for the decompression parameters. The compression parameters for standard data sets stored in the CMPR data base may not be overridden by users (the command-line qualifiers will be ignored) so the compression technique for a standard data set is under configuration control.

If a file is compressed from the DCL command level, a system-unique temporary file is used to store the data. If the compression is successful (a shorter file is created), the original file is replaced by the temporary file. All permanent attributes except the record-length are retained. Since the file name, version, extension, creation and modification dates are unchanged, the archive catalog need not be updated. Since the modification date is unchanged, VMS BACKUP software will not restore any offlined version of the same file, reducing the offline storage volume.

8. Compression Method Specification

The compression methods so far envisaged consider a packet of successive records ("chunk') as an image to be compressed. The methods may require parameters (such as a range), positional information (matrix partition) and a specification of the number of records in the buffer. Although non-optimal, each block ,delimited by a specified field offset range, is constrained to a fixed number of records in the buffer. The block may be scanned in column order ("transposed"), row or rectangular image and variable-length output is reformatted and re-aligned to fixed-length records. An optional list of reference filenames may be provided and a list of floating point parameters may be required.

The following generic compression schemes are provided :

Field : data fields are compressed by re-quantization.

Scanline : data in a "horizontal" or "vertical" range of scanlines is compressed by methods which consider correlations between adjacent data elements. The FULL vertical (time-series) scanline is compressed.

Block : data in a non-overlapping, multiple range of offsets is compressed by methods which consider the correlations between neighboring elements. The operators may be causal, acausal or semi-causal in scanline order.

9. Compressed Data Record Structure

The existing data management system is based on a fixed-length record structure with field offsets defined in an RDL file. The record-length and connect-time file attributes are stored in a database under Configuration Management control. Since many data compression methods generate variable-length records, it was necessary to devise a scheme permitting full random access without wasting storage on record filler bytes.

Since the time code and pixel address label fields are strictly monotonically increasing (except for certain data sets not destined for compression) this may be achieved as follows:

Figure 3 demonstrates the separate compression of field offset ranges for "packets" of fixed-length records with fixed-length output records supporting full random access by time code and pixel address. The status byte indicates whether the record is compressed or not and the "lookback" word points to the beginning of the output record. The shaded areas denote successive samples of data in pre-defined offset ranges. Subrecords are broken across the record boundary with the label fields deferred to the beginning of the next output record. In this manner, if the search finds a label value the "lookback" field refers to the start of the compressed data associated with that label. The result is that these fields are never split across record boundaries and no space is wasted.

A restriction is placed on the length of an output record that it must not exceed the length of the "lookback" field plus the length of the status field. The output record length is constrained to always exceed this value so no input record may span more than two output records. Any output record that exceeds this limit is transmitted in clear codes. Any compressed file larger than the original is transmitted in uncompressed form.

10. Random Access to Compressed Data.

The efficient search for matching time codes in large time-ordered files requires the insertion of an internal time code index list at predefined records in the uncompressed file. When a file is opened the first index list is read into virtual memory. If the desired record is not in the decompressed buffer, the bounding time codes are searched for in the index list to minimize the I/O. If the time codes are not found in the current index list, the next list is read into memory. The search uses a hunt and locate method, where the initial record is predicted from the average compression factor for the file, determined from the compressed file size and the number of uncompressed records multiplied by the uncompressed record length stored in the archive catalog. The exponential search is carried out until the time codes are bracketed when a binary search is used to locate the exact compressed record. The compressed record buffer is searched linearly for the matching time codes. The reduction in I/O by using index lists leads to an order of magnitude improvement in search time.

The search for matching pixel addresses in a sky map proceeds similarly except that an index file pointing to the first logical (uncompressed) record under a pixel is already available. Two lists of corresponding logical and physical (compressed) record numbers are stored in the file. In both cases, the index lists are highly compressible.

11. Compression Algorithms

The initial algorithmic toolbox will contain range quantization, run-length coding and zero suppression methods. The range quantization is an approximate method currently used by DIRBE which recognizes the sentinel values flagging noisy data.

Planned subsequent development includes nested Chebyshev polynomials (smoothly-variable data), a modified Huffman code, the Haar Transform followed by quadtree bit plane encoding, variants of the Lempel-Ziv-Welch substitution schemes with static codebooks, stochastic models such as the Autoregressive Integrated Moving Average (ARIMA) schemes and tree-structured Vector Quantization based on static codebooks. Since the data distribution is almost stationary with time, a static codebook may be stored on in memory for codebook-based algorithms. Usage of the Vector Quantization algorithms will depend on available resources and will probably be restricted to static archives.

12. Worked Example

The DIRBE experiment was operated with cryogenic cooling for 41 weeks, creating 80MB per day for a total of 5.5GB processed data.

Clearly, this RDL was devised with each field carefully specified for scientific usage and it is not necessary to make minimalist assumptions about the nature of the data. This RDL specifies a mixture of scientific and engineering data and some fields must be transmitted in clear codes (search labels), exactly (flags), approximately (photometry) or are noisy and hence incompressible (e.g. pixel subposition).

The records are 140 bytes long and in quadsphere sky map format [10]. The label field is the "Pixel_No" which is referenced explicitly in the user software as a pixel-number-offset argument to the access software. There are 16 floating-point photometric bands.

Direct usage of "Unix-compress" leads to a compression factor of 25% which takes several hours to compress one sky map on a workstation.

DIRBE has already decided that a logarithmic range compression scheme which sentinelizes glitchy data (flagged in a previous pipeline process) is sufficient to convert the floating-point photometry to 16 bit integers on a field-by-field basis. Further compression may be achieved particularly for data which are not glitchy (Glitch_Flags) or taken in a particle radiation zone (Radiation_Cont). This represents about 75% of the data. This compressible data may be vector quantized with a suitable codebook derived (perhaps) by the Linde-Buzo-Gray algorithm based on a training set extracted from a typical daily file. A normalized codebook would be the most flexible. At best, this approach would yield ~ 2 bytes per array of highly-correlated photometry bands.

The ratio of the daily photometry to the annual average value under a pixel is expected to have reduced dynamic range and be even more compressible. The "pixel_no" fields and the ancillary angles between the DIRBE boresight and celestial objects which vary slowly under a pixel are ~50% compressible via a Modified Huffman Code in vertical scan mode. The overall compression factor is about 50%.

In another example, one FIRAS facility accesses time-ordered data via extensive keyed-read operations which involve searches which currently create the largest single network load.
The files are approximately 16MB consisting of 8 byte time codes together with ~ 10 000 bytes of data. Each search step (there are typically about 6 per keyed read) reads the whole record to locate the time code. The compressed data which contains the internal time code list may be searched ~ 30 times faster as the total I/O is reduced to 10% of its original value.

13. Validation and Testing

Software quality has been assured by regression testing in an independent environment to ensure that goals of functionality, accuracy and performance have been met. Code inspection has been used to ensure the robustness and maintainability of the code and documentation.

The in-house validation team will provide quality assurance for the compressed data products using the same formal project accuracy requirements as for original data.

Tests of file migration to/from all available media (including 4mm and 8 mm magnetic tape, magnetic and optical disks and 9-track tape) indicate that the compressed data files are fully compatible with VMS BACKUP and COPY software and that the project-specific data migration software facility is effective with compressed data.

14. Summary and Conclusions

A general approach incorporating scientific knowledge seems appropriate for the Space and Earth Science Data Compression application. Inline data compression techniques developed for the COBE project may help the project achieve its goals and be useful to other workers in this growing field.

15. Recommendations for Future Development

Compression functions should be specified at the same time the data sets are defined. An optimal implementation may consider the data as a linked list of object classes for each data field which specify overloaded (de)compression functions invoked in the constructor for each class.

A Data Compression Designer Expert System could capture the knowledge of domain experts and recommend appropriate functions.

Acknowledgements

The COBE data analysis is managed by the Goddard Space Flight Center for NASA's Office of Space Science and Applications.
COBE is a team effort and more than 1000 individuals have contributed to its success.

References

[1] J. A. Storer, M. Cohn (eds.), Proc. Data Compression Conf. 1992, held at Snowbird, Utah, April 24-27, 1992.

[2] N. W. Boggess et al., "The COBE Mission;Its Design and Performance Two Years after Launch", Ap. J. 1 October, 1992.

[3] J. C. Mather et al., "The COBE FIRAS Measurement of the Cosmic Microwave Background Spectrum", Ap. J. , 1993 (in preparation).

[4] M. G. Hauser et al.,"The Diffuse Infrared Background : COBE and other results",in "After the First Three Minutes", AIP Conf. Proc. 222, 1991.

[5] J. Weiland et al., "COBE/Diffuse Infrared Background Experiment (DIRBE) Observations of the Milky Way Galactic Bulge", Ap. J. Lett., 1993 (in preparation).

[6] T. Sodroski et al., "Large Scale Physical Conditions in the Interstellar Medium from DIRBE Observations", Ap. J. Lett., 1993 (in preparation).

[7] E. L. Wright et al., "Preliminary Spectral Observations of the Galaxy with a 7 degree beam by the Cosmic Background Explorer (COBE)", Ap. J., 21 October, 1991.

[8] W. Spiesmann et al.,"Near and Far Infrared Observations of Interplanetary Dust Bands from COBE", Ap. J. Lett., 1993 (in preparation).

[9] E. S. Cheng, "COBE: The Software", Proc. First Astronomical Data Analysis Software and Systems Conf., held at Tucson, Arizona, Nov 8-11, 1993.

[10] I. Freedman, A. C. Raugh, E. S. Cheng, "The COBE Astronomical Databases", Proc. First Astronomical Data Analysis Software and Systems Conf., op. cit.

[11] R. A. White, S. W. Stemwedel,"The Quadrilateralized Spherical Cube and Quad-Tree for All Sky Data", Proc. First Astronomical Data Analysis Software and Systems Conf., op.cit.

[12] I. M. O'Neill, R. E. Laubscher,"Extended Studies of a Quadrilateralized Spherical Cube Earth Data Base", NEPRF Tech. Rep. 3-76 (CSC).

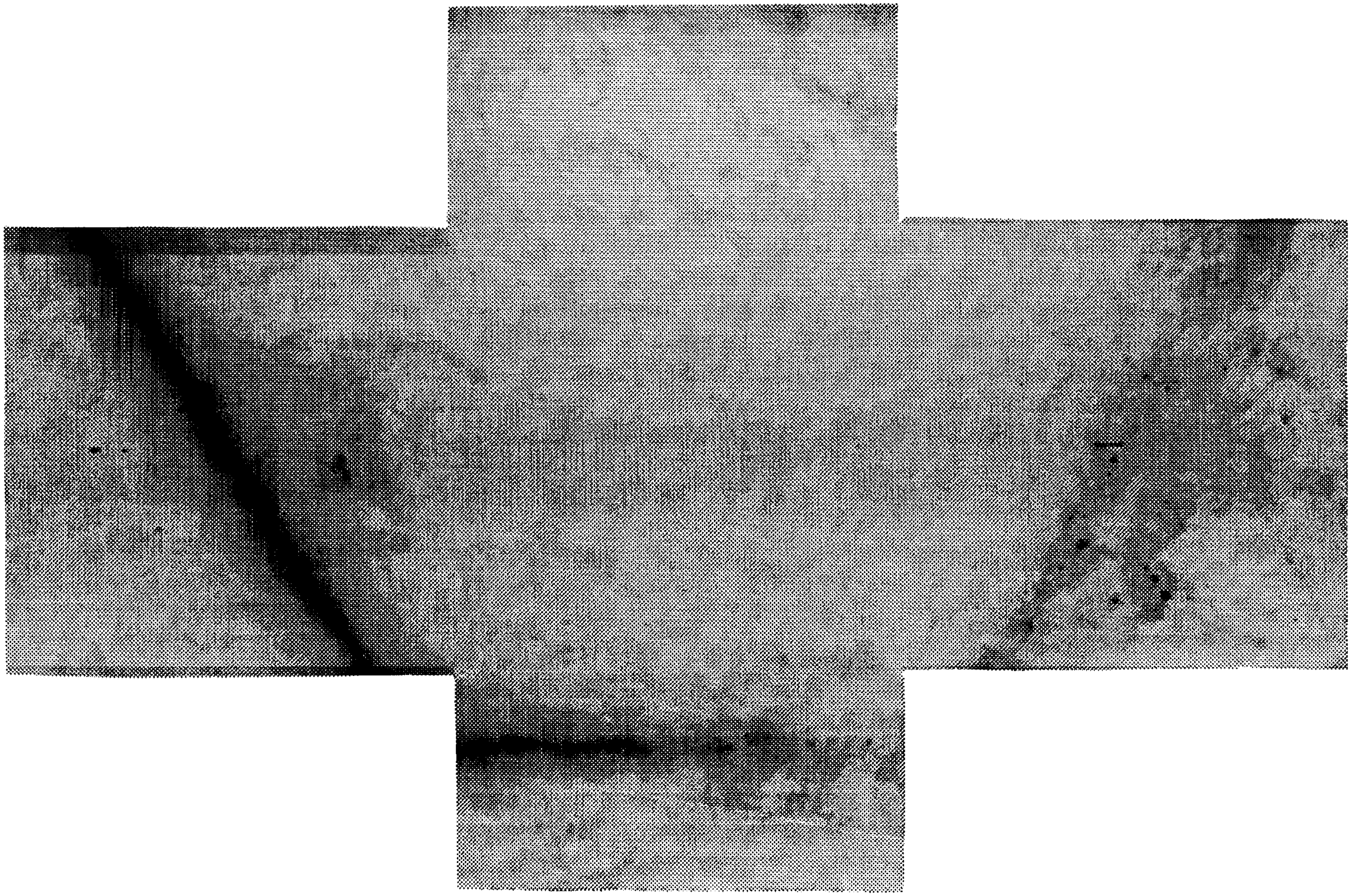
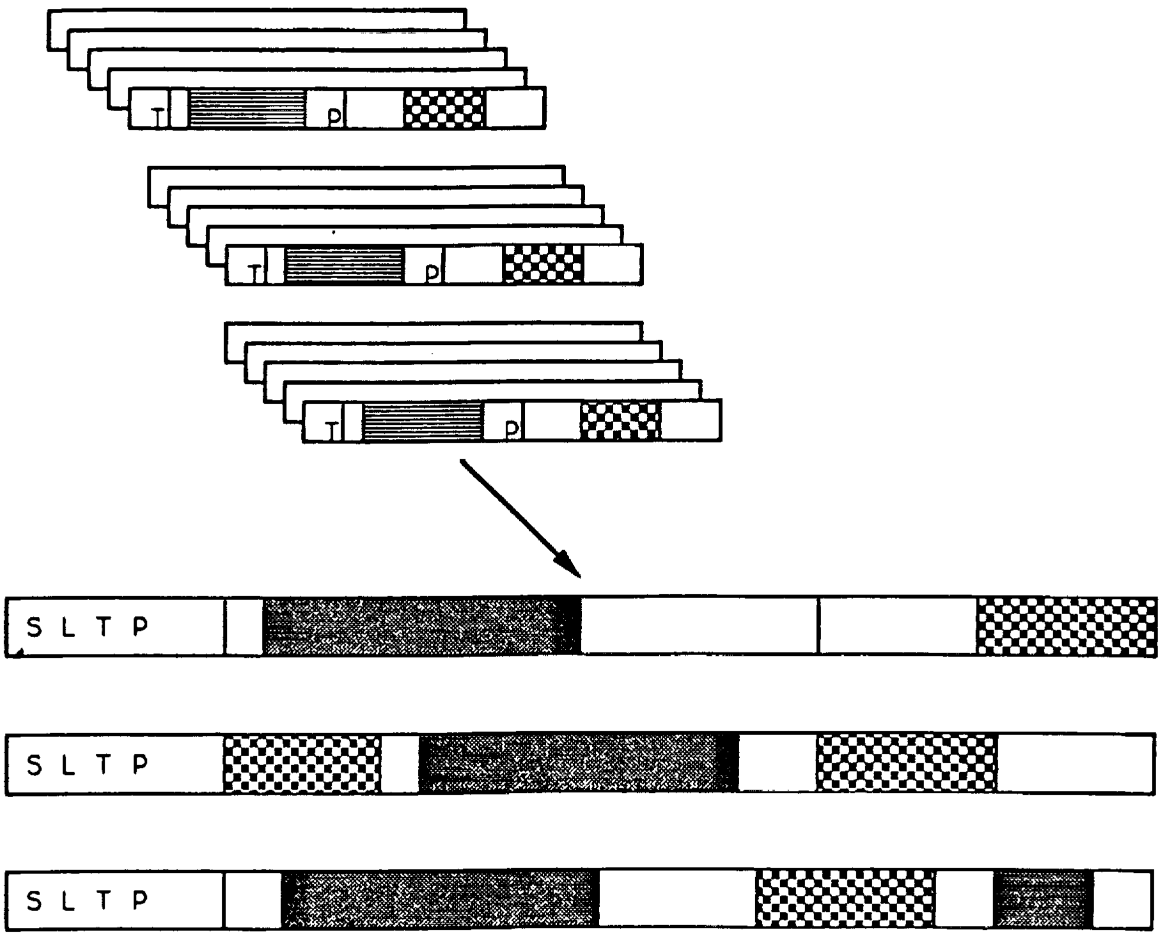


Figure 1. Annual Average 100 μm map projected on quadsphere.

```
RECORD BCI_CIRSSM BCI_CIRSSM  ! Complete IRS Sky Maps

Offset  Length  Description
------  ------  -----------
   0       8      SCALAR Time          /ADT!Time of middle of observation.
   8       4      SCALAR Pixel_no      /LONG!Pixel number of observation
                                            !
  12      64      ARRAY  Photometry    /FLOAT/DIM=16!Detector observations

  76       1      SCALAR Approach_vector /BYTEU!Forward looking =  1
                                              !Backward looking = 2
                                              (referenced to SC velocity)
  77       1      SCALAR Pixel_subpos  /BYTEU!Sub-pixel containing DIRBE LOS
  78       4      SCALAR Next_obs      /LONG!Pixel number of next observation
  82       4      SCALAR Prev_obs      /LONG!                 previous

  86       4      ARRAY Sun_re_BS      /WORD/DIM=2
                                       !Word 1:Solar elongation.
                                       !Word 2: Relative azimuth of sun
  90       2      SCALAR SC_Axis_re_Zenith /WORD!Angle between COBE -X axis
                                              !and the zenith vector
  92       2      SCALAR BS_re_Zenith /WORD!Angle between DIRBE boresight and
  94       2      SCALAR BS_re_Horiz   /WORD!Angle between earth horizon and
                                           ! DIRBE boresight.
  96       2      SCALAR SC_Axis_re_Vel /WORD!Angle of COBE -X axis relative
                                            !to velocity vector
  98       2      SCALAR BS_re_Vel     /WORD !Angle between DIRBE boresight
                                            !and S/C velocity vector
 100       2      SCALAR Azimuth_re_Vel /WORD
 102       6      ARRAY Attack_vector  /WORD/DIM=3
 108       2      SCALAR FOV_Azimuth   /WORD
 110       2      SCALAR Longitude     /WORD
 112       2      SCALAR Latitude      /WORD
 114       2      SCALAR Altitude      /WORD
 116       3      ARRAY  Mag_Field     /BYTE/DIM=3
 119       4      ARRAY Moon_re_BS     /WORD/DIM=2
 123       2      SCALAR Sun_Moon_Angle /WORD
 125       2      SCALAR Moon_Distance /WORD
 127       4      ARRAY Jupiter_re_BS  /WORD/DIM=2

 131       4      ARRAY Earth_light_cont /WORD/DIM=2

 135       1      SCALAR Pixel_subsubpos  /BYTEU

 136       1      SCALAR Radiation_cont /BYTEU
 137       2      SCALAR Glitch_Flags   /WORDU
 139       1      SCALAR ATT_Flags      /BYTEU
 140            END_RECORD

TOTAL LENGTH OF RECORD:    140 BYTES
TOTAL NUMBER OF FIELDS:     29
```

Figure 2. Record Definition Language for DIRBE Daily File.

FRAMEWORK



Separate compression of field offset ranges for "packets" of fixed-length records with fixed-length output records supporting full random access by time-tag and pixel number. The status byte indicates whether the record is compressed or not and the "lookback" word points to the beginning of the output record. The shaded areas denote successive samples of data in pre-defined offset ranges. The notation "S L T P" denotes status, lookback , time-tag and pixel number.

Figure 3. Separate compression of field offset ranges.