THE DIGITIZED SKY SURVEY

Discs 1–61

The Southern Hemisphere

Mean Data Compression Factor: 10

These CD-ROMs were prepared by the Catalogs and Surveys Branch of the Space Telescope Science Institute. The following individuals made significant contributions: Doris Daou, Jesse B. Doggett, Ian N. Evans, Jeffrey J. E. Hayes, Victoria G. Laidler, Barry M. Lasker, Brian J. McLean, Michael Meakes, Flavio J. Méndez, Marc Postman, Michael M. Shara, Conrad R. Sturch, and Richard L. White.

Production of these discs was funded partially by U. S. Government grant NAG W-2166.

These discs were mastered to ISO 9660 standard, level 1.

The images on these discs are based on photographic data obtained using The UK Schmidt Telescope. The UK Schmidt Telescope was operated by the Royal Observatory Edinburgh, with funding from the UK Science and Engineering Research Council, until 1988 June, and thereafter by the Anglo-Australian Observatory. Original plate material is copyright © the Royal Observatory Edinburgh and the Anglo-Australian Observatory. The plates were processed into the present compressed digital form with their permission. The *Digitized Sky Survey* was produced at the Space Telescope Science Institute (ST ScI) under U. S. Government grant NAG W-2166. *Investigators using these scans are requested to include the above acknowledgments in any publications.*

Victoria G. Laidler and Conrad R. Sturch are with the Astronomy Programs, Computer Sciences Corporation at the ST ScI.

Artwork by John Godfrey and Dave Paradise.

1. INTRODUCTION

The Space Telescope Science Institute (ST ScI) digitized Schmidt survey plates covering the entire sky to obtain the image data needed for construction of the *Guide Star Catalog* (GSC) and to pursue a number of related research programs. Copies of the GSC are made available to the scientific community and general public through the ST ScI and the Astronomical Society of the Pacific (ASP), respectively. The digitized versions of the plates are of great utility in astronomical research as well, but distribution of the scans has previously been impractical because of the massive volume of data involved (a total of about 600 Gbytes). However, the H-transform wavelet compression technique now makes such a distribution feasible. With funding from NASA Headquarters, compression of 1479 digitized images covering the entire sky began in June 1992. Part I in a series of CD-ROM publications comprises 61 discs, and contains a complete imaging survey of the southern hemisphere (plate centers $\delta \leq 0°$) constructed largely from the SERC Southern Sky Survey and the SERC J Equatorial extension (see Table III of Lasker *et al.* 1990, hereafter Paper I). These are deep (3600 s) IIIa-J exposures obtained through a GG 395 filter, except for 94 short (1200 s) V-band exposures mostly at low galactic latitudes ($|b| \leq 15°$) plus 2 plates covering the Large Magellanic Cloud and 2 very short (300 s) V-band exposures, each centered on one of the Magellanic Clouds (see Paper I). The digitized images have been compressed by a factor of 10, on average, and are extremely faithful to the original data (*cf.* §7.3). The 61 discs contain a total of 896 compressed, digitized Schmidt images, an astrometric calibration database, and data access software. Part II of this series is the 583 E (red) plates with centers $\delta \geq +6°$ from the 1950–55 epoch Palomar sky survey.

Questions, comments, and suggestions regarding this publication may be directed to

asp@bkyast.berkeley.edu	(Internet), or
(415) 337-5205	(FAX), or paper mail to

Astronomical Society of the Pacific
Dept. 10X
390 Ashton Avenue
San Francisco, CA 94112-1787
U. S. A.

A detailed presentation of the technical and scientific aspects of these compressed data is being prepared (Postman *et al.* 1994); the information contained in this booklet provides a brief summary of the most important issues.

2. THE SCANS

The plates were digitized using the ST ScI scanning microdensitometers, which are described in Paper I. A pixel size of $25\,\mu$m with a $50\,\mu$m apodized aperture was used throughout. The pixel data are on a photographic density scale, with density 5 being $2^{15}-1$, right adjusted in a 16 bit word. Therefore, to convert data values (DN) to photographic density (γ) use the expression

$$\gamma = DN/6553.4.$$

The nominal $25\,\mu$m pixel size is actually $25.284\,\mu$m ($1''\!.70$ at the Schmidt plate scale of $67''\!.2\,\mathrm{mm}^{-1}$), which is a natural unit in the system of the HeNe laser used for positional measurement.

The southern scans have undergone extensive quality assurance checks. Two artifacts that can be introduced during digitization are chopping (misalignment of odd and even pixels within individual columns) and shearing (misalignment of odd and even scan rows). Algorithms to measure these effects are based on the symmetries of the appropriate correlations. Fourier methods are then used to achieve the necessary repairs. The deshearing is a simple application of the shifting theorem, while dechopping is a special case of the "interlace" problem (Bracewell 1965). Of the 896 scans on discs 1–61, 14 required deshearing and 34 required dechopping. In general, repaired scans are indistinguishable from good scans with small or negligible digitization artifacts.

3. DISC ORGANIZATION

Each disc has a root directory which contains one or more files and several directory trees. In the root directory of all 61 discs is a file identified COPYRGHT.;1 containing the copyright notice for the disc, and directories identified for each plate on the disc. The latter contain the compressed image data. In addition, in the root directory of disc 61, there is also a file identified README.TXT;1 containing a copy of this introduction, and the following two additional directories.

- HEADERS Contains astrometric calibration header files, one for each plate, and other required calibration files.
- SOFTWARE Contains GetImage software for accessing the compressed data, including the files identified GETIMAGE.BCK;1 (OpenVMS backup saveset of GetImage) and GETIMAGE.TAR;1 (UNIX tar file of GetImage). There is also a directory tree GETIMAGE that contains an unpacked version of the software. The top directory of the GETIMAGE directory tree contains files to compile and run GetImage under OpenVMS and UNIX systems, and a directory identified SOURCE that contains the source files for GetImage.

The contents of the directory tree GETIMAGE are discussed in §4. The contents of the directories which contain the compressed data and the calibration files are described in §5.

Each disc has a volume label of the form USA_AURA_STSI_DSS1_*nnnn*, where *nnnn* is the right justified, zero filled disc number. The set of discs does not comprise an ISO 9660 volume set.

4. SOFTWARE

4.1 The GetImage Program

A program called GetImage for reading the compressed images is provided in the `SOFTWARE` directory tree on disc 61. The software is provided in three forms: as a backup saveset for OpenVMS systems (`GETIMAGE.BCK;1`), as a tar file for UNIX systems (`GETIMAGE.TAR;1`), and in an unpacked form in the directory tree `GETIMAGE`. The backup saveset and tar files include installation instructions for their respective systems. The unpacked form includes the installation instructions for both OpenVMS and UNIX systems. We recommend that the backup saveset or tar file be used to extract the software for the appropriate system. Instructions for unpacking GetImage are given below.

GetImage takes as input the equatorial coordinates for the center of an astronomical field and the size of the rectangular field. The user will be prompted to mount the appropriate disc for the field, if it is not already mounted. The image corresponding to the requested field will then be extracted. The input may be specified either interactively or from a file. The extracted images can be written as either FITS files (the default) or as GEIS (`hhh/hhd`) images.

GetImage requires about 8 Mbytes of disk space to run. This is because simultaneous access to the plate image headers and the plate image data is required. Thus, the header files must be copied from the `HEADERS` directory on disc 61 to magnetic disk. Procedures for copying the header files are included with the installation instructions provided with the software.

GetImage has been successfully built and run on Digital Equipment Corporation VAX and AXP computers running the OpenVMS operating system, and on Sun Microsystems IPC and Model 2 SPARCstations running the SunOS UNIX operating system. The typical time required to extract a $10'$ field is about 10 s on a DEC 3000 M400 workstation with 64 Mbytes of RAM, about 30 s on a DEC VAXstation 3100 with 24 Mbytes of RAM, and about 50 s on a SUN Sparcstation IPC with 24 Mbytes of RAM.

It should be possible to use the software with slight modifications on other machines and other operating systems. In the top directory of the `GETIMAGE` directory tree, the file identified `OSDEPEND.TXT;1` lists the places in the source of GetImage that are operating system dependent. Other system-dependencies may be introduced if the machine uses other than 32-bit integers for C `int` variables.

4.2 Installation on OpenVMS Systems

To install GetImage on OpenVMS systems:

1) Mount disc 61 as appropriate for your system. (See your system manager if you are unsure about how to do this.)
2) Set your default directory to the desired parent directory for GetImage. (Note: There must be at least 8 Mbytes of free disk space there.)
3) Issue the following `BACKUP` command, where `CDROM` is the device on which disc 61 is mounted:

   ```
   $ BACKUP CDROM:[SOFTWARE]GETIMAGE.BCK/SAVE_SET [...]
   ```
4) Set your default directory to the directory `[.GETIMAGE]` that was created with the `BACKUP` command.

5) Follow the instructions in the file `INSTALL.VMS` to compile GetImage and copy the plate image header files to magnetic disk.

6) Read the file `README.TXT` for information on how to use GetImage.

4.3 Installation on UNIX Systems

To install GetImage on UNIX systems:

1) Mount disc 61 as appropriate for your system. (See your system manager if you are unsure about how to do this.)

2) Change directory to the desired parent directory for GetImage. (Note: There must be at least 8 Mbytes of free disk space there.)

3) Issue the following `tar` command, where `/cdrom` is the mount point where disc 61 is mounted:

```
% tar -xf /cdrom/software/getimage.tar
```

4) Change directory to the directory `getimage` that was created with the `tar` command.

5) Follow the instructions in the file `install.unx` to compile GetImage and copy the plate image header files to magnetic disk.

6) Read the file `readme.txt` for information on how to use GetImage.

5. THE DATA AND ASTROMETRIC SOLUTIONS

The original plate images are composed of 14000×13999 pixels. To allow fast access to any portion of the plates, the images were divided into blocks of 500×500 pixels (500×499 for the last row) that were compressed separately. The directories with compressed data therefore contain 784 compressed subimage files identified as *plate*.`00;1`, *plate*.`01;1`, ..., *plate*.`09;1`, *plate*.`0A;1`, ..., *plate*.`0R;1`, *plate*.`10;1`, ..., *plate*.`RR;1`. The GetImage image extraction software allows any arbitrary section of a plate to be extracted; only the blocks that are needed for the requested section are uncompressed.

In the `HEADERS` directory on disc 61, there are header files for all the plate images contained on discs 1–61. These header files, identified as *plate*.`HHH;1`, are text files containing a FITS-like description of the plate scans. Keywords other than those for standard FITS headers describe the details of the plate scans, including a polynomial solution for mapping X, Y pixel coordinates on the plate to right ascension and declination. The polynomial form of the astrometric solutions is identical to that adopted for the GSC (versions 1.0 and 1.1). The solutions for plates that were not desheared or dechopped are indeed the GSC V1.0/1.1 solutions. A new astrometric solution was computed for the 48 plates that were processed through our deshearing or dechopping algorithms.

To compute an equatorial position (J2000) from pixel coordinates X, Y (computed with respect to the origin of the full 14000×13999 pixel image) one must first convert X, Y to units of mm from the plate center x, y,

$$x = (x_c - p_x X)/1000,$$
$$y = (p_y Y - y_c)/1000,$$

where x_c and y_c are the plate center coordinates in μm (assigned to keywords PPO3 and PPO6, respectively, in the FITS header) and p_x and p_y are the x and y dimensions of a pixel in μm (assigned to keywords XPIXELSZ and YPIXELSZ, respectively, in the FITS header). One then constructs the standard coordinates ξ, η,

$$\begin{aligned}\xi &= a_1 x + a_2 y + a_3 + a_4 x^2 + a_5 xy + a_6 y^2 \\ &\quad + a_7 (x^2 + y^2) + a_8 x^3 + a_9 x^2 y + a_{10} x y^2 \\ &\quad + a_{11} y^3 + a_{12} x (x^2 + y^2) + a_{13} x (x^2 + y^2)^2, \\ \eta &= b_1 y + b_2 x + b_3 + b_4 y^2 + b_5 xy + b_6 x^2 \\ &\quad + b_7 (x^2 + y^2) + b_8 y^3 + b_9 x y^2 + b_{10} x^2 y \\ &\quad + b_{11} x^3 + b_{12} y (x^2 + y^2) + b_{13} y (x^2 + y^2)^2,\end{aligned}$$

where $a_1, \ldots, a_{13}$ are assigned to keywords AMDX1, ..., AMDX13 and $b_1, \ldots,$ b_{13} are assigned to keywords AMDY1, ..., AMDY13. The keywords AMDX14, ..., AMDX20 and AMDY14, ..., AMDY20 are not currently used. The standard coordinates, as computed above, will be in units of arcseconds. Finally, the J2000 celestial coordinates α, δ (in radians) are computed from the standard coordinates as follows:

$$\begin{aligned}\alpha &= \arctan\left[(\xi \,/\, \cos\delta_c) \,/\, (1 - \eta \tan\delta_c)\right] + \alpha_c, \\ \delta &= \arctan\left\{\left[(\eta + \tan\delta_c)\cos(\alpha - \alpha_c)\right] / \left[1 - \eta \tan\delta_c\right]\right\},\end{aligned}$$

where α_c is the plate center right ascension (assigned to keywords PLTRAH, PLTRAM, and PLTRAS) and δ_c is the plate center declination (assigned to keywords PLTDECSN, PLTDECD, PLTDECM, and PLTDECS). The variables α_c, δ_c, ξ, and η must be converted to radians before using the above expressions! If $\alpha < 0$, one must add 2π to its value to get the correct right ascension.

The origin convention adopted in generating the astrometric solutions is that the X, Y coordinates of the lower left hand corner of the lower left hand pixel in a 14000×13999 pixel image are $(1.0, 1.0)$. Therefore, the center of the lower left hand pixel is $(1.5, 1.5)$. This is contrary to the FITS standard (and the expectations of some popular image analysis packages) which define the center coordinates of the origin pixel to be $(1.0, 1.0)$. If using software that expects pixel centers to have integral coordinate values, a $(+0.5, +0.5)$ offset should be added to the measured X, Y coordinates prior to computing celestial coordinates. Failure to do so could result in a $\sim 1\farcs2$ position error.

One also must assure that any image display software properly sets the absolute values of the origin coordinates for the particular subimage being processed. Some image display packages generate only relative coordinates (*ie.* the origin is always $(0, 0)$ or $(1, 1)$, and, consequently, gross position errors could be introduced. The proper X, Y coordinates of the lower left hand corner of the lower left hand pixel for any given subimage are stored in the keywords `CNPIX1` and `CNPIX2`, respectively.

The above information is provided for those who wish to compute object coordinates once a subimage has been uncompressed. We suggest examining the functions `readhr` (found in the file `header.c` of the GetImage software) and `amdpos` (found in the file `astrmcal.c` of the GetImage software) to see how the keywords are used in a real algorithm. The code described by Russell *et al.* (1990) was used to generate the polynomials that support these utilities. Photometric solutions have not been provided with Part I of the *Digitized Sky Survey*; however, a photometric calibration will be provided as Part III of this series.

6. NETWORK ACCESS POLICY

Users of stand-alone computing systems with *no* remote access capability may access the ST ScI *Digitized Sky Survey* CD-ROMs without a license as long as they note and comply fully with the copyright limitations given in this booklet and in the image header files. For the purposes of this policy, remote access is defined as any method of electronic data transfer which allows these data to be copied from the host machine to remote nodes. Such methods include, but are not restricted to, serial lines, local area and wide area networks.

Users wishing to provide remote access to these data may do so as long as they (1) note and comply fully with the copyright limitations stated in this booklet and in the image header files, *and* (2) obtain an appropriate license by contacting:

Office of Contract & Business Services
Attn: Digitized Sky Survey
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, MD 21218
U. S. A.

The license agreement for non-profit users will place specifications on the identification of the data and its sources and there will be no additional fees.

All users of these data are asked to acknowledge both the original data sources (given in the image headers) and the ST ScI digitization and compression.

7. IMAGE COMPRESSION

7.1 The H-transform

The image compression technique we are using is based on the H-transform (Fritze *et al.* 1977; see also Richter 1978 and Capaccioli *et al.* 1988). The H-transform is a two-dimensional generalization of the Haar transform (Haar 1910). The H-transform is calculated for an image of size $2^N \times 2^N$ as follows:

- Divide the image up into blocks of 2×2 pixels. Call the 4 pixels in a block a_{00}, a_{10}, a_{01}, and a_{11}.
- For each block compute 4 coefficients:

$$\begin{aligned} h_0 &= (a_{11} + a_{10} + a_{01} + a_{00})/2, \\ h_x &= (a_{11} + a_{10} - a_{01} - a_{00})/2, \\ h_y &= (a_{11} - a_{10} + a_{01} - a_{00})/2, \\ h_c &= (a_{11} - a_{10} - a_{01} + a_{00})/2. \end{aligned}$$

- Construct a $2^{N-1} \times 2^{N-1}$ image from the h_0 values for each 2×2 block. Divide that image up into 2×2 blocks and repeat the above calculation. Repeat this process N times, reducing the image in size by a factor of 2 at each step, until only one h_0 value remains.

This calculation can be inverted easily to recover the original image from its transform. The transform is exactly reversible using integer arithmetic if one does not divide by 2 for the first set of coefficients. The extension of the definition of the transform so that it can be computed for non-square images that do not have sides which are powers of 2 is straightforward. The H-transform can be performed in place in memory and is very fast to compute, requiring only about $4M^2$ (integer) additions for a $M \times M$ image.

If the image is nearly noiseless, the H-transform is somewhat easier to compress than the original image because the differences of adjacent pixels (as computed in the H-transform) tend to be smaller than the original pixel values for smooth images. Consequently, fewer bits are required to store the values of the H-transform coefficients than are required for the original image. For very smooth images the pixel values may be constant over large regions, leading to transform coefficients which are zero over large areas.

Noisy images still do not compress well when transformed, though. Suppose there is noise σ in each pixel of the original image. Then from simple propagation of errors, the noise in each of the H-transform coefficients is also σ. To compress noisy images, divide each coefficient by $S\sigma$, where $S \sim 1$ is chosen according to how much loss is acceptable. This reduces the noise in the transform to $0.5/S$, so that large portions of the transform are zero (or nearly zero) and the transform is highly compressible.

Why is this better than simply thresholding the original image? If we simply divide the image by σ then we lose all information on objects that are within 1σ of sky in a single pixel, but that are detectable by averaging a block of pixels. On the other hand, in dividing the H-transform by σ, we preserve the information on any object that is detectable by summing a block of pixels! The quantized H-transform preserves the mean of the image for every block of pixels having a mean significantly different than that of neighboring blocks of pixels.

7.2 Coding the H-transform

The quantized H-transform has a rather peculiar structure. Not only are large areas of the transform image zero, but the non-zero values are concentrated strongly in the lower-order coefficients. The best approach we have found to code the coefficient values efficiently is quadtree coding of each bitplane of the transform array. Quadtree coding has been used for many purposes (see Samet 1984 for a review); the particular form we are using was suggested by Huang and Bijaoui (1991) for image compression.

- Divide the bitplane up into 4 quadrants. For each quadrant code a '1' if there are any 1-bits in the quadrant, else code a '0'.
- Subdivide each quadrant which is not all zero into 4 more pieces and code them similarly. Continue until one is down to the level of individual pixels.

This coding (which Huang and Bijaoui call "hierarchic 4-bit one" coding) is obviously very well suited to the H-transform image because successively lower orders of the H-transform coefficients are located in successively divided quadrants of the image.

We follow the quadtree coding with a fixed Huffman coding that uses 3 bits for quadtree values which are common (*eg.* 0001, 0010, 0100, and 1000) and uses 4 or 5 bits for less common values. This reduces the final compressed file size by about 10% at little computational cost. Slightly better compression can be achieved by following quadtree coding with arithmetic coding (Witten, Neal, and Cleary 1987), but the CPU costs of arithmetic coding are not, in our view, justified for 3–4% better compression.

For completely random bitplanes, quadtree coding can actually use more storage than simply writing the bitplane directly; in that case we just dump the bitplane with no coding.

7.3 Astrometric and Photometric Properties of Compressed Images

We have conducted experiments to study the degradation of astrometry and photometry on the compressed images compared to the original images (White, Postman, and Lattanzi 1992). Even the most highly compressed images have very good photometric properties for both point sources and extended sources; indeed, photometry of extended objects can be improved by the adaptive filtering of the H-transform (Cappacioli *et al.* 1988). Astrometry is hardly affected by the compression for modest compression factors (up to about a factor of 20 for our digitized Schmidt plates), but does begin to degrade for images with higher compression factors.

8. ACKNOWLEDGMENTS

We acknowledge gratefully grant number NAG W-2166 from the Science Operations Branch of NASA headquarters which funded partially this image compression project, and thank Guenter Riegler for his encouragement and support. We sincerely thank Riccardo Giacconi for his support over the years in making our work on the *Guide Star Catalog* useful both for the *Hubble Space Telescope* project and for other science, and to Bob Williams for his critical support in the final stages of the CD-ROM publication. We also thank Rich Casciano, Harry Feinstein, Dick Rossi, and Bob Ramseth for administrative and management support. Finally, we acknowledge Bob Havlen and the rest of the staff of the Astronomical Society of the Pacific for providing distribution services.

Special thanks go to the Royal Observatory Edinburgh and the Anglo-Australian Observatory for permission to distribute these scans of plates obtained with The UK Schmidt Telescope.

9. REFERENCES

Bracewell, R. 1965, in *The Fourier Transform and its Applications*, McGraw Hill, New York, chapter 10.

Capaccioli, M., Held, E. V., Lorenz, H., Richter, G. M., and Ziener, R. 1988, *Astron. Nachr.*, **309**, 69.

Fritze, K., Lange, M., Möstl, G., Oleak, H., and Richter, G. M. 1977, *Astron. Nachr.*, **298**, 189.

Haar, A. 1910, *Math. Ann.*, **69**, 331.

Huang, L., and Bijaoui, A. 1991, *Experimental Astron.*, **1**, 311.

Lasker, B. M., Sturch, C. R., McLean, B. J., Russell, J. L., Jenkner, H., and Shara, M. 1990, *Astron. J.*, **99**, 2019 (Paper I).

Postman, M., White, R. L., Lasker, B. M., Hayes, J., Doggett, J., Meakes, M., and Mendez, F., 1994, in preparation.

Richter, G. M. 1978, *Astron. Nachr.*, **299**, 283.

Russell, J. L, Lasker, B. M., McLean, B. J., Sturch, C. R., and Jenkner, H. 1990, *Astron. J.*, **99**, 2059.

Samet, H. 1984, *ACM Computing Surveys*, **16**, 187.

White, R. L., Postman, M., and Lattanzi, M. 1992, in *Digitized Optical Sky Surveys*, H. T. MacGillivray and E. B. Thomson (eds.), Kluwer Academic Publishers, Netherlands, pp. 167–175.

Witten, I. H., Neal, R. M., and Cleary, J. G. 1987, *Commun. ACM*, **30**, 520.