SZIP: Data Compression on the Sphere

User Manual

J. D. McEwen[1*] and D. M. Eyers[2†]

[1]*Astrophysics Group, Cavendish Laboratory, J. J. Thomson Avenue, Cambridge CB3 0HE, UK*

[2]*University of Cambridge Computer Laboratory, J. J. Thomson Avenue, Cambridge CB3 0FD, UK*

SZIP is a utility for compressing data defined on the sphere. Large data-sets that are measured or defined inherently on the sphere arise in a range of applications. Examples include, astronomical observations that are made on the celestial sphere, such as the cosmic microwave background (CMB), geophysics data and environmental illumination maps and reflectance functions used in computer graphics. Technological advances in observational instrumentation and improvements in computing power are resulting in significant increases in the size of data-sets defined on the sphere (hereafter we refer to a data-set defined on the sphere as a *data-sphere*). For example, current and forthcoming observations of the anisotropies of the CMB are of considerable size. Recent observations made by the NASA WMAP satellite contain approximately three mega-pixels, while the forthcoming ESA Planck mission will generate data-spheres with approximately fifty mega-pixels. The efficient and accurate compression of data on the sphere is therefore becoming increasingly important for both the dissemination and storage of data.

Motivated by the requirement for a data-sphere compression algorithm defined on a constant latitude pixelisation of the sphere, and a publicly available tool to compress such data, we have developed wavelet-based compression algorithms for data defined on the HEALPix pixelisation scheme. These algorithms are implemented in SZIP, which we make publicly available[1] for academic use (commercial use is absolutely prohibited). We are driven primarily by the need to compress CMB data, hence the adoption of the HEALPix scheme (the pixelisation scheme used currently to store and distribute these data). Wavelet transforms are expected to perform well in the energy compression stage of the compression algorithm, thus we have adopted a Haar wavelet transform on the sphere for this stage. Both lossless and lossy compression algorithms are developed. The "lossless" algorithm is lossless to a user specified numerical precision due to a quantisation stage introduced to improve compression ratio. As one increases the precision parameter, lossless compression is achieved in the limit (while trading off compression ratio). The lossy algorithm introduces additional error in a controlled manner and, by allowing a small degradation to the fidelity of the compressed data, significantly greater compression ratios can be attained. Optional run length encoding (RLE) can also be performed, which usually proves beneficial for lossy compression. For more information on the compression algorithms implemented in SZIP and an evaluation of their performance please see our related academic paper [1]. In this manual we describe how to install, use and test your copy of SZIP.

1 Installation

For various licensing reasons it is not possible to release SZIP open source. We therefore release binary executables of SZIP for a number of different platforms, including Windows, Mac OS X and Linux (if your platform is not supported please get in touch and we will do our best to add support for this platform soon). The installation of SZIP therefore simply involves saving the appropriate executable to your disk. SZIP is a command line application and must be called from within a terminal (in Mac OS X and Linux) or from a DOS prompt in Windows. Read on for details on how to use SZIP.

*mcewen@mrao.cam.ac.uk

†david.eyers@cl.cam.ac.uk

[1]http://www.szip.org.uk

2 Usage

SZIP must be run from the command line (from within a terminal in Linux or Mac OS X; from a DOS prompt in Windows). Currently, support is only provided to compress HEALPix pixelised data-spheres, although in future we may add support for other pixelisations of the sphere (please contact us if this would be useful for you). Moreover, only HEALPix data-spheres in the NESTED format can be compressed by SZIP; data-spheres in the RING format are not accepted. We cannot currently support data-spheres in RING format since we are not able to release SZIP as open source and so cannot use HEALPix code to convert between RING and NESTED formats. We may add functionality to SZIP in future to support RING ordered data-spheres but hope that potential users can convert their RING ordered data to a NESTED format (using, say, HEALPix) before compressing their data.

The SZIP utility itself provides basic usage help by printing a list and brief description of all command line options. To view this list simply run SZIP with the `--help` option. For example, running

```
>> ./szip --help
```

will print the following usage information:

```
SZIP data sphere compression utility
(c) 2007 Jason McEwen and Dave Eyers

Usage:
 -h  --help             Prints this help
 -c  --compress         Compress input file
 -x  --decompress       Extract compressed input file
 -i  --input <infile>   Input file
 -o  --output <outfile> Output file
 -l  --level <num>      Compression level [default=1]
 -p  --sigfig <num>     Precision (no. of significant figures) [default=5]
 -r  --rle              Run length encoding
 -y  --lossy <factor>   Loss factor (percentage to keep) [default=100]
 -b  --nbins <num>      Number of bins used to compute threshold in lossy compression [default=100]
 -q  --quiet            Surpress comments and warnings
 -t  --time             Run profiler
 -v  --version          Print version number
```

Either the short one character command line options or the longer options (both shown above) can be used interchangeably. We now go on to explain each of the command line options shown in the usage list above.

- `--help`

  Print usage information.

- `--compress`

  Run SZIP in compress mode to compress an input .fits data-sphere (the input file must be a HEALPix data-sphere in the NESTED ordering scheme).

- `--decompress`

  Run SZIP in decompress mode to decompress an input .sip compressed file.

- `--input <infile>`

  Specify the input file as <infile>. If running SZIP in compress mode this file must have a .fits extension; if running in decompress mode this file must have a .szip extension.

- `--output <outfile>`

  Specify the output file as <outfile>. If running SZIP in compress mode this file must have a .szip extension; if running in decompress mode this file must have a .fits extension. If no output file is specified then it will be set automatically with the same name as the input file but with the appropriate extension.

- `--level <num>`

  Specify the wavelet analysis depth to perform Haar wavelet transform down to as num. By default this value is set to 1, which should be suitable for most applications. For further detail see our related paper [1], where <num> is denoted by the variable J0.

- `--sigfig <num>`

  Specify the precision parameter to use in the quantisation stage as <num>. By default this value is set to 5. Increasing this value improves the fidelity of compression but trades off compression ratio performance. This parameter actually specifies the number of significant figures to retain in the representation of the wavelet detail coefficients.

- `--rle`

  Perform run length encoding (RLE) in addition to the usual Huffman encoding. It usually proves beneficial to include RLE when performing lossy compression but not for lossless compression (since RLE introduces an additional coding overhead)

- `--lossy <factor>`

  Perform lossy compression and specify the loss factor as <factor>. The loss factor determines the proportion of detail coefficients to retain in the lossy compression and may range from 0 to 100 percent. By default this factor is set to 100 (i.e. all detail coefficients are retained) and lossless compression is performed.

- `--nbins <num>`

  Specify by <num> the number of bins to use when constructing the histogram used to determine the threshold level required to retain the appropriate portion of detail coefficients in lossy compression.

- `--quiet`

  Suppress comments and warnings that are otherwise printed when running SZIP.

- `--time`

  Run timing profiler when running SZIP.

- `--version`

  Print SZIP version number.

We now give a couple of examples of running SZIP with various command line options. To compress the data-sphere `sky.fits` with default parameter values run

```
>> ./szip --compress --input sky.fits
```

or equivalently

```
>> ./szip -c -i sky.fits
```

These commands will produce the compressed file `sky.szip`. To decompress this file run

```
>> ./szip --decompress --input sky.szip --output sky_recon.fits
```

or equivalently

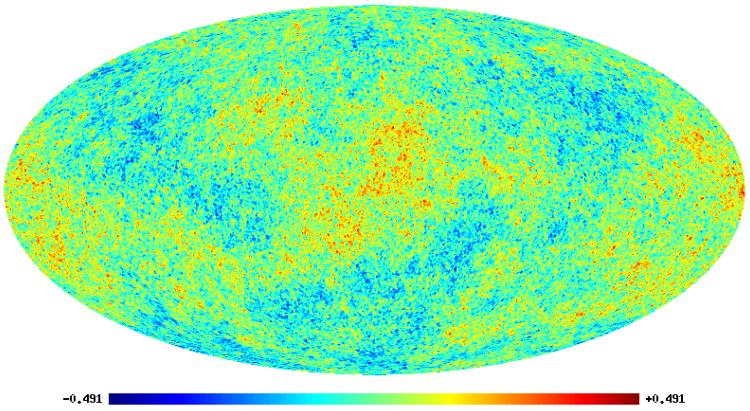
```
>> ./szip -x -i sky.szip -o sky_recon.fits
```

No parameter values are used in the decompression stage so these last two commands can always be used to decompress SZIP files (of course, with the appropriate filenames). To compress `sky.fits` with a precision parameter of 4, using lossy compression to retain only 10 percent of wavelet detail coefficients and to add RLE run

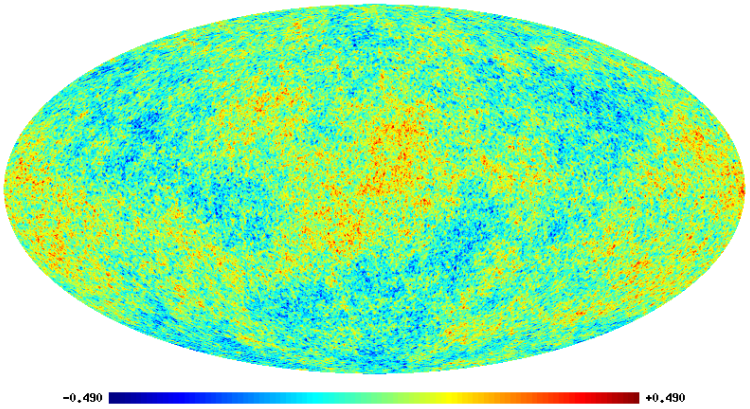
```
>> ./szip --compress --input sky.fits --output sky_p4_y10_rle.szip --sigfig 4 --rle --lossy 10
```

or equivalently

```
>> ./szip -c -i sky.fits -o sky_p4_y10_rle.szip -p 4 -r -y 10
```



(a) Original data



(b) Decompressed data

Figure 1: Simulated CMB data before and after compression-decompression.

3 Testing

To test your copy of SZIP we make a simulated CMB map available on the SZIP web site: `gcmb001_n256_nest.fits`. An image of this map is shown in Fig. 1 (a). Compress and decompress these data by running

```
>> ./szip -c -i gcmb001_n256_nest.fits
```

followed by

```
>> ./szip -x -i gcmb001_n256_nest.szip -o gcmb001_n256_nest_recon.fits
```

An image of the reconstructed map is shown in Fig. 1 (b). This should match an image of the decompressed map you obtain by running your local copy of SZIP. Notice the small error between Fig. 1 (a) and (b), apparent from the different limits on the colour bars. This error arises since compression has been performed with a precision parameter of 5 (the default value). By increasing the precision parameter the reconstruction error can be reduced. A detailed study of the trade off between the fidelity of the compressed data against compression ratio is performed in our related paper [1]. Furthermore, we also discuss the cosmological implications of these types of error on CMB data.

Hopefully your SZIP installation is working successfully. We hope you enjoy SZIP and that it proves useful for your particular application. Any feedback on the current version or suggestions for improvements would be warmly received. If you do use SZIP in academic work that results in publication please reference the SZIP home page[2] and our paper [1].

[2] `http://www.szip.org.uk`

References

[1] J. D. McEwen, Y. Wiaux and D. M. Eyers, Data compression on the sphere, Astron. & Astrophys., 531, A98, 2011