CSIRO PUBLISHING
www.publish.csiro.au/journals/pasa *Publications of the Astronomical Society of Australia*, 2005, **22**, 1–5

Random Numbers from Astronomical Imaging

Kevin A. Pimbblet[A,C] *and Michael Bulmer*[B]

[A] Department of Physics, University of Queensland, Brisbane QLD 4072, Australia
[B] Department of Mathematics, University of Queensland, Brisbane QLD 4072, Australia
[C] Corresponding author. Email: pimbblet@physics.uq.edu.au

Received 2004 April 30, accepted 2004 August 16

Abstract: This article describes a method to turn astronomical imaging into a random number generator by using the positions of incident cosmic rays and hot pixels to generate bit streams. We subject the resultant bit streams to a battery of standard benchmark statistical tests for randomness and show that these bit streams are statistically the same as a perfect random bit stream. Strategies for improving and building upon this method are outlined.

Keywords: methods: statistical — techniques: image processing — techniques: miscellaneous

1 Introduction

Random numbers are of importance to many sub-fields of science. In observational astrophysics they are required for diverse uses (Meurers 1969) such as testing for sub-structure in galaxy clusters (Dressler & Shectman 1988) and Monte Carlo background correction techniques (Pimbblet et al. 2002). In cryptography, the generation of secure passwords and cryptographic keys is paramount to communication being immune from eavesdropping. They are also used in selecting winning numbers for lotteries including the selection of Premium Bonds in the United Kingdom (`www.nsandi.com/products/pb/`). Large Monte Carlo computations, however, remain the primary driver of intensive searches for truly random number generators (e.g. Ferrenberg, Landau, & Wong 1992; James 1990).

For many purposes we would essentially like to have a long bit stream consisting of 1s and 0s. Each bit in the stream should be independently generated with equal probability of being a 1 or 0. Therefore as the length of the stream, n, tends toward infinity, the expectation value of any individual bit being either 1 or 0 is $1/2$. The traditional method of obtaining such a stream is to use a pseudo-random number generator (PRNG; e.g. Press et al. 1992). PRNGs typically rely upon the input of a 'seed' quantity which is then processed using numerical and logical operations to give a stream of random bits. Whilst such PRNGs are probably sufficient for most (minor) types of applications, they are clearly predictable if the initial seed is known. This makes PRNGs highly inappropriate for Monte Carlo-like calculations (González & Pino 1999).

A truly random number generator (RNG) should possess qualities that make the bits unpredictable. The obvious sources of RNGs are those that possess large amounts of entropy or chaos (Vavriv 2003; Gleeson 2002; González & Pino 1999). Examples include radioactively decaying sources (e.g. HotBits; `www.fourmilab.ch/hotbits/`), electrical noise from a semiconductor diode, and thermal noise.

An overlooked and potentially large source of random numbers is to be found in astronomical imaging. Imaging at a telescope will inevitably produce unwanted cosmetic features such as cosmic ray events, satellite trails, and seeing effects — blurring due to the movement of the atmosphere (in the case of ground-based telescopes). It is precisely these features (and in particular cosmic rays) which potentially make astronomical imaging a good RNG.

This article presents an assessment of astronomical imaging data as a source for a RNG. In Section 2, we demonstrate how it is possible to generate a stream of random bits from a single astronomical image. Examples of such bit streams are examined in Section 3 using a battery of statistical tests to evaluate their randomness. Our findings are summarised in Section 4.

2 Generating Random Bits

Assuming that one is in possession of a sample of astronomical images that possess cosmic ray events we can proceed to obtain a bit stream from them by following the procedure outlined in Figure 1. We detail the individual steps below.

Our aim is to detect the locations of any cosmic ray or 'hot spot' (pixel values that are significantly greater than their local neighbours) in the pixel distribution. For this experiment, we use single-shot exposures of 300 to 600 s from non-overlapping wide-field observations consisting of 2×4k pixels from Pimbblet and Drinkwater (2004) and their on-going follow-up observations[1]. Firstly, we use the IRAF (`iraf.noao.edu/`) task COSMICRAYS with

[1] It is also unnecessary to pre-process these images with flat fields, for example, as all we are interested in are the locations of hot pixels. Indeed, our testing has shown that a raw image produces just an equally random bit stream as a post-processed one does. One problem that is encountered

10.1071/AS04043 1323-3580/05/01001

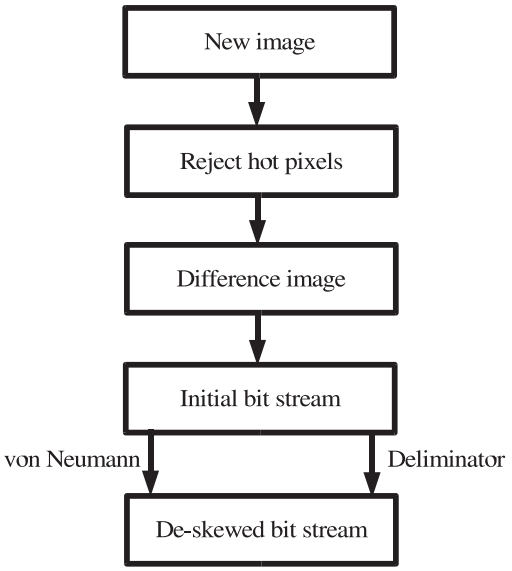

Figure 1 Overview of the processes required to generate a random bit stream from an initial astronomical image. Using a Dell Precision workstation 530 machine for analysing an initial image of 2×4k pixels, the generation of an initial bit stream takes about 50 s whilst the de-skewing requires about 30 s (for either the von Neumann or deliminator method). The average rate of random bit production is approximately 2500 bits per second (bps). Depending upon the amount of discarded bits, this figure can range from as low as 1000 bps up to 4000 bps.

default parameters to remove the cosmic rays from the original image. Then, using IMARITH, we subtract the cosmic ray free imaging from the original to create a difference image in which there should be only cosmic rays (Figure 2). Inevitably, this technique will identify not only true cosmic rays but also anomalously hot pixels from the distribution. To turn the difference image into a bit stream, we sequentially examine the contents of each pixel in turn, row by row, column by column. Pixels with a value of zero in the difference image translate into a 0 for the bit stream whilst those with values greater than zero (the hot pixels) become 1.

The fraction of pixels identified as cosmic rays (and hot pixels) using this method is typically 2–3% for our exposures. Clearly, there exist more 0s in the bit stream than there are 1s. Moreover, there are distinct 'holes' in the hot pixel distribution of the difference image where legitimate objects occurred in the original image (the galaxy in Figure 2). So, whilst there are random events in our bit stream, it is highly skewed toward 0s.

2.1 De-skewing

To turn our bit stream into a uniformly random distribution, it is necessary to de-skew it (an 'entropy distillation process'; Rukhin et al. 2001). Here we adopt and investigate two common methods of de-skewing. The first is

is the presence of bad pixels, which always occur in the same place on a CCD. These should be removed with the FIXPIX (or similar) task before proceeding.

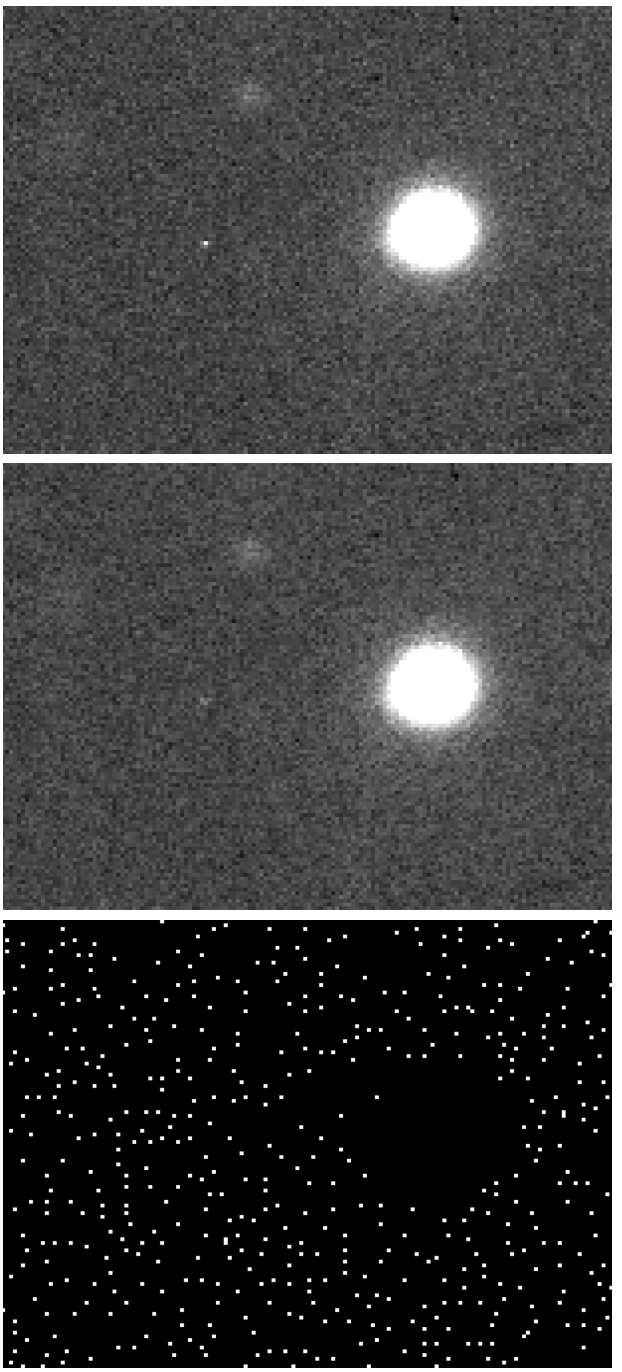

Figure 2 Example of the image processing method. Top: a subsection of the original image measuring 613×480 pixels. Middle: the cosmic ray rejected version of the image. Bottom: the difference image. Note how the obvious cosmic ray (left of centre) is rejected along with a host of other relatively 'hot' pixels. Real objects, meanwhile, leave an obvious hole in the difference image which requires de-skewing to generate a random bit stream.

that of von Neumann (1963). We read the bit stream generated from the imaging as a sequence of non-overlapping pairs. The pairs are then transformed into a new bit stream according to the scheme presented in Table 1. This scheme removes all biases in the original bit stream at the expense of drastically reducing the overall size of the original, as it removes the long sequences of 0s associated with

https://doi.org/10.1071/AS04043 Published online by Cambridge University Press

legitimate objects (Table 2). The typical reduction for our imaging is in the range 85–98%, although this is a highly variable parameter.

The second method is to use the hot pixels (or groups of hot pixels; 1s) as deliminators between long streams of 0s. The length of non-overlapping pairs of long streams of 0s are then compared to each other to generate a 1 or a 0, depending if the first stream is longer than the second or vice versa. If the lengths are equal, nothing is appended to the new bit stream. An example of how both of these methods work is illustrated in Table 2. The clear disadvantage of the delimination method is that a much smaller bit stream is produced than for the von Neumann method.

3 Evaluating the Randomness

In truly random bit stream, each bit should be generated with probability $1/2$ of producing either a 0 or 1. Further, each bit should be generated independently of any other bit in the bit stream. One should not, therefore, be able to predict the value of a given bit by examining the values of the bits generated prior to it in the bit stream. These conditions define an ideal, truly random bit stream, and we will use them to test our random bits against.

To evaluate the randomness of our bit stream, we subject it to a battery of benchmark statistical tests. The tests we use are a selection of those devised by Random Number Generation and Testing collaboration of the National Institute of Standards and Technology (NIST; Gaithersburg, MD, USA; `csrc.nist.gov/rng/index.html`). The NIST statistical test suite source code is freely available from their web site.

For each test, the software formulates a specific null hypothesis (H_0) and alternative hypothesis (H_1). We will specify that H_0 is the hypothesis that our bit stream is random and that H_1 is the hypothesis that it is non-random. To accept or reject H_0, one determines a test statistic and compares it to a critical value chosen to be in the tails of a theoretical reference distribution of the test statistic. The possible outcomes of the statistical testing are illustrated in Table 3. The probability of obtaining a Type I error is therefore the level of significance of a test (see Rukhin et al. 2001), which we set at a level of 0.01 for this work.

The software determines a P-value for each test: the probability that a *perfect* RNG would produce a bit stream that is *less* random than the bit stream that we test (i.e. a P-value of zero denotes a bit stream that is certainly non-random). Therefore to reject the null hypothesis H_0 (and hence fail the test) at a 99% confidence level would require a P-value <0.01. Clearly, the P-value only assesses the relative incidence of Type I errors. What it does not describe is the probability that a non-random number generator could produce a sequence of numbers at least as random as our bit stream that is being tested (a Type II error; Rukhin et al. 2001).

Here, we briefly describe each test (the tests and the statistics behind them are described in much more detail in Rukhin et al. 2001) and summarise the results in Table 4.

If our bit stream is random, then the number of 1s and 0s overall and in any part (i.e. any sub-sequence) of it should be approximately the same. Therefore the first test is to examine the frequency of 1s and 0s in the bit stream. In the second test, these frequencies are re-computed for sub-sequence blocks of length M (see also Knuth 1981; Pitman 1993).

Table 1. Scheme for the von Neumann (1963) de-skewing method. The original bit stream from the imaging is read in as a sequence of non-overlapping pairs ('Input Pair' column). The output for the new bit stream is then given in the 'Output' column. Where 'Null' is indicated, nothing is appended to the new bit stream

Input pair	Output
00	Null
11	Null
01	0
10	1

Table 3. Possible configuration of the conclusions of any of the given statistical hypothesis tests (Rukhin et al. 2001)

True situation	Result	
	Accept H_0	Reject H_0
H_0 true	Correct	Type I error
H_1 true	Type II error	Correct

Table 2. A comparison of how the de-skewed bit stream is generated using the von Neumann (1963) and deliminator methods

	Bit stream
Pixel distribution	`0010000100000100101001000100100011000110100 01`
von Neumann pairings	`++-++-++-++-++-++-++-++-++-++-++-++-++-++-++`
von Neumann de-skewed	`..1...0.......1.0.0...1...1.0...0.1.....0...0.`
Deliminator pairings	`++++++++----------++++++-------++++++++++------`
Deliminator de-skewed	`0.......1.........0....1................0.....`

https://doi.org/10.1071/AS04043 Published online by Cambridge University Press

Next, we test the number of runs in the sequence, where a run is defined as an uninterrupted sequence of identical bits. This tests if the bit stream oscillates with sufficient celerity between 1s and 0s. We follow this test by a similar one that evaluates the longest run of 1s in the sequence to determine if it is the same as would be expected for a random distribution. If there is an irregularity in the longest run length of 1s, then this will also be reflected in the longest run length of 0s; hence we only test the longest run length of 1s.

Table 4. The proportion of 100 bit streams of length $n = 10^6$ that passed each statistical test (critical P-value 0.01). The minimum pass rate in order for our sequence to be considered random is approximately 0.96 for each statistical test (see Rukhin et al. 2001)

Test	Proportion passed		Notes
	von Neumann	Deliminator	
Frequency	0.98	1.00	
Block frequency	0.98	1.00	$M = 1000$
Runs	0.97	0.99	
Longest run	0.99	0.97	
Rank	1.00	0.99	
Cusum	0.97	0.99	
DFT	1.00	1.00	

To test for any linear dependence of sub-strings of fixed length within the original sequence, the rank of disjoint sub-matrices is examined. This method is described in more detail by Marsaglia in the DIEHARD statistical tests (`stat.fsu.edu/~geo/diehard.html`).

We can also consider the bit stream as a random walk and hence test the maximal excursion from zero for the cumulative sum (cusum) of adjusted digits ($+1$, -1) in our bit stream or sub-sequence therein (Revesz 1990). For a random sequence, cusum should be near zero. Finally, by performing a discrete Fourier transform (DFT), we can look for periodic features in our bit stream that would indicate a lack of randomness.

Table 4 shows that both variants of the de-skewing method are sufficient to pass all of the standard tests outlined above by at least the minimum pass rate (Rukhin et al. 2001). We can further assess the validity of our conclusion by examining the distribution of P-values, which for a random sequence should be approximately uniform. For this, we re-run our experiment but use 1000 bit streams of length 10^5 (since 100 bit streams comprise a relatively small sample). All the tests outlined above (Table 4) are passed once again and we display the distribution of P-values for these in Figure 3. All of the distributions of P-values approximate uniformity very well. To test the uniformity of the distributions, we can use a

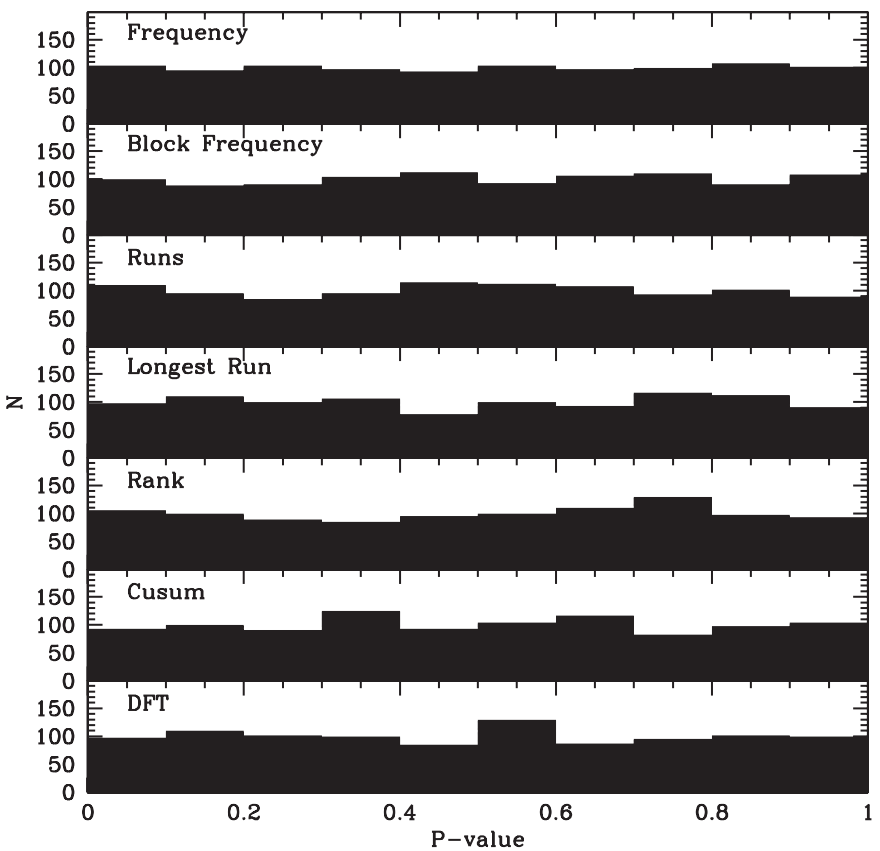

Figure 3 Histograms of the P-value distributions arising from applying the seven statistical tests from Table 4 to 1000 bit streams of length 10^5. All of the distributions are approximately uniform.

https://doi.org/10.1071/AS04043 Published online by Cambridge University Press

χ^2 test and a determination of a P-value that corresponds to the goodness-of-fit distributional test of the P-values (a so-called 'P-value of P-values'; Rukhin et al. 2001). The χ^2 statistic is simply:

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - 100)^2}{100} \quad (1)$$

where F_i is the number of P-values in bin i of Figure 3. The P-value of P-values is then the complemented incomplete gamma function:

$$1 - \Gamma(a, z) \quad (2)$$

where we set $a = 9/2$ and $z = \chi^2/2$ (see Rukhin et al. 2001). This yields a mean χ^2 value for the distributions in Figure 3 of 9.9 ± 2.8 whilst $1 - \Gamma(9/2, \chi^2/2) = 0.357$. Since $1 - \Gamma(9/2, \chi^2/2)$ is much larger than (say) 0.0001, we can consider the distributions to be uniformly spread.

We note, however, that by altering the size of the bit stream downward (say $n = 10^5$), we have been able to cause the von Neumann de-skewed variant to fail the runs test. This emphasises the fact that these tests need to be carried out on a large bit stream sample (at least $n \geq 10^6$).

4 Summary

We have described how astronomical imaging can be used as a true RNG by application of simple cosmic ray rejection algorithms. Although we throw away a large fraction of our original data through the de-skewing methods, we have shown that resultant bit stream is sufficiently random to pass modern tests for randomness.

The tests that we applied are only a selection of the NIST statistical test suite. There are more within this suite and certainly more beyond (e.g. Tu & Fischbach 2003; Ballesteros & Martín-Mayor 1998; Knuth 1981). We have therefore looked at applying more complex tests for randomness, as detailed in the NIST test suite (e.g. non-overlapping template matching, etc.). We find that these additional tests are readily passed by both de-skewed variants of our bit streams.

Several improvements to our methodology can potentially be made. We are looking at different de-skewing techniques to improve the test statistics. For example, the von Neumann method can be used twice (or more) on the bit stream generated from the image. The resulting proportions of bit streams that pass the statistical tests in Table 4 increases fractionally as a result, but not significantly. Our next step is to attempt to create a web interface where it will be possible to download random numbers in real time using this method. This could be accomplished by using the international network of continuous cameras (concams; Nemiroff & Rafert 1999). The concams have the virtues that one does not require the sky to be dark locally and the images are freely available to the public.

The imaging used in this work is at optical wavelengths (specifically B, V, R, and I-bands). It may be interesting to examine how the test results varied with other parts of the electromagnetic spectrum, if at all. The imaging is also non-overlapping. If the concams constitute a valid RNG, then it is worthwhile to confirm that images of the same area of sky produce independent random bit streams, which should be the case as we are only considering the incidence of hot pixels (i.e. cosmic rays).

Accessory Materials

One sequence of 10^6 bits (von Neumann de-skewed; approximately 1.1 Mb in size) is available from the author or, until January 2010, from *Publications of the Astronomical Society of Australia*. Please note that this bit stream is only one small part of the much larger sample used to generate the results presented in this work. Your mileage may vary!

Acknowledgments

This work has made use of the NIST statistical test suite. We wish to warmly thank NIST for allowing public access to this code. We also thank the anonymous referee for a very positive and thorough review that has improved the clarity of this work. K.A.P. acknowledges an EPSA University of Queensland Fellowship. The imaging used in this work is from the WFI instrument used on both the Anglo-Australian Telescope and Siding Spring Observatory 40″ Telescope and we thank both observatories for the time allocated to us.

References

Ballesteros, H. G., & Martín-Mayor, V. 1998, PhRvE, 58, 6787
Dressler, A., & Shectman, S. A. 1988, AJ, 95, 985
Ferrenberg, A. M., Landau, D. P., & Wong, Y. J. 1992, PhRvL, 69, 3382
Gleeson, J. T. 2002, ApPhL, 81, 1949
González, J. A., & Pino, R. 1999, CoPhC, 120, 109
James, F. 1990, CoPhC, 60, 329
Knuth, D. E. 1981, The Art of Computer Programming. Vol. 2: Seminumerical Algorithms (Reading, MA: Addison-Wesley)
Meurers, J. 1969, A&A, 3, 354
Nemiroff, R. J., & Rafert, J. B. 1999, PASP, 111, 886
Pimbblet, K. A., & Drinkwater, M. J. 2004, MNRAS, 347, 137
Pimbblet, K. A., Smail, I., Kodama, T., Couch, W. J., Edge, A. C., Zabludoff, A. I., & O'Hely, E. 2002, MNRAS, 331, 333
Pitman, J. 1993, Probability (New York, NY: Springer)
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in FORTRAN. The Art of Scientific Computing (Cambridge: CUP)
Revesz, P. 1990, Random Walk in Random and Non-Random Environments (Singapore: World Scientific)
Rukhin, A., et al. 2001, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (Gaithersburg, MD: NIST) (csrc.nist.gov/rng/SP800-22b.pdf)
Tu, S., & Fischbach, E. 2003, PhRvE, 67, 016113
Vavriv, D. D. 2003, Proc. Experimental Chaos (Melville, NY: AIP) 676, 373
von Neumann, J. 1963, Various Techniques Used in Connection with Random Digits, von Neumann's Collected Works, Vol. 5 (Oxford: Pergamon)

https://doi.org/10.1071/AS04043 Published online by Cambridge University Press