IJDC | *Peer-Reviewed Paper*

Recovery of Heritage Software Stored on Magnetic Tape for Commodore Microcomputers

Denise de Vries, Craig Harrington
Flinders University, South Australia

Abstract

Digital games make up a significant but little known chapter in the history of the moving image in Australia and New Zealand. Beginning in the early 1980s, the Australasian software industry developed a remarkable record of content creation. The 'Play It Again' project is conducting research into the largely unknown histories of 1980s game development in Australia and New Zealand, ensuring that local titles make it into national collections and are documented and preserved, enabling the public to once again play these games.

Microcomputers from the 1980s made extensive use of compact audio cassettes to distribute software as an inexpensive alternative to the floppy disk technology available at the time. Media from this era are at risk of degradation and are rapidly approaching the end of their lifespan. As hardware platforms and peripheral devices become obsolete, access to the data for future scholars and other interested parties becomes more difficult. In this article, we present a case study, wherein we investigate the issues involved in making digital copies with a view to the long term preservation of these software artefacts.

A video game title stored on standard compact cassette for Commodore's popular VIC-20 machine, 'Dinky Kong' by Mark Sibley was recorded using both inexpensive amateur and professional playback equipment. The audio files obtained were processed using freely available software, alongside a customised decoder written in MATLAB and Perl. The resulting image files were found to be playable using an emulator. More importantly, the integrity of the data itself was verified, by making use of error detection features inbuilt to the Commodore tape format, which is described in detail.

Issues influencing the quality of the recovered image files, such as the bit rate of the digital recording, are discussed. The phenomenon of audio dropout on magnetic tape is shown be of some concern, however there exist signal processing techniques to compensate for such errors.

The end result of the imaging process was a file compatible with a popular Commodore VIC-20 emulator, the integrity of which was verified by using inbuilt checksums.

Received 14th October 2015 | *Revision received* 14th December 2016 | *Accepted* 14th December 2016

Correspondence should be addressed to Denise de Vries, CSEM, GPO Box 2100, Adelaide SA 5001
Email: denise.deVries@flinders.edu.au

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

http://dx.doi.org/10.2218/ijdc.v11i2.386
DOI: 10.2218/ijdc.v11i2.386

Introduction

A number of early home microcomputers made use of standard compact audio cassettes to store software and data. In comparison to floppy disk drives, the playback/recording hardware was inexpensive and the media were readily available. With the advent of music stored on compact disk and now consumed mostly by digital download, audio cassettes are becoming increasingly more difficult to work with due to obsolescence of the hardware.

In this paper, we describe the issues encountered in creating an image of the Commodore VIC-20 game, 'Dinky Kong'. This title was coded by Mark Sibley as a teenager, locally distributed in New Zealand and is of historical significance due to the fact that Sibley has subsequently created significant software works, such as Blitz BASIC for the Commodore Amiga and more recently, the Monkey X programming language (Sibley, 2014).

All magnetic media are susceptible to incurring damage over a period of years, which is informally known as 'bit rot'. This can be either due to demagnetisation of the data laden layer or degradation of the media itself. In the case of magnetic tape, the latter case is more common: the binders which hold together the layers of the tape are prone to disintegrate over time (Gibson, 1997). Thus, it is desirable to preserve the information before the data are lost forever.

The DC2N device (Fraia, 2006) offers a hardware solution to the problem of creating digital copies of the data on tape. Image files of Commodore (or Sinclair) tapes are able to be produced on non-volatile memory card using an original Commodore tape drive directly connected to the DC2N. That is, there is no need for either a host system or a legacy Commodore machine in the process, although the device itself can function as a replacement for the Commodore tape drive, if desired. Unfortunately, this device has only been produced in somewhat limited quantities and was unavailable for purchase, leading to the need to investigate alternative methods for creating tape images.

Guttenbrunner et al. (2009) presented methodologies for preserving and extracting data from cassettes for a Philips G7400. In their work, having access to the original equipment enabled them to re-engineer the audio waveform, as well as all data formats, and write a tool to migrate the data to non-obsolete formats.

In this article, the process of creating a tape image is described. Conventional audio cassette playback equipment was used to capture the data which was subsequently transformed into in a format compatible with a popular Commodore emulator. The main contributions of this work can be summarised as follows:

- Description of Commodore VIC-20 tape format,
- Capture of legacy data without proprietary hardware,
- Description of the errors encountered during data capture,
- Creation of an error-free TAP file image.

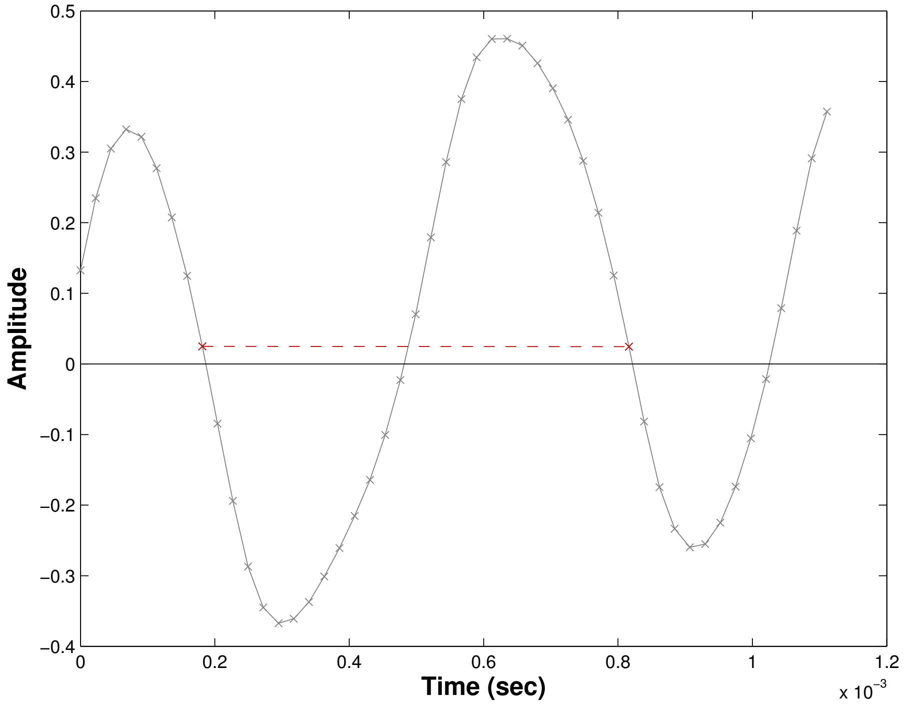

Figure 1. Plot of tape waveform sampled at 44100 Hz, illustrating approximate duration of a long pulse.

Background

Magnetic Tape Data Format

The tape format used by Commodore's PET, VIC-20 and 64 makes use of standard compact audio cassettes with tones within normal range of human hearing. Specifically, the pulse width modulation (PWM) scheme employed allows for data to be decoded at a rate of 300 baud using the standard loaders present in Commodore's operating system (or KERNAL) (Commodore Business Machines Inc., 1984). This technique has the advantage of being relatively robust to presence of noise, simple to implement and most importantly, insensitive to slight variations in the playback speed of the magnetic tape (Georgiou, 1984).

The beginning of a pulse is represented by a falling edge zero crossing: where the waveform changes from positive amplitude to negative amplitude. The end point of the pulse is the next such crossing. The distance between these two points may be considered in order to classify the pulse as either a short, medium or long, as shown in Figure 1.

The Commodore tape format for an executable file consists of a lead-in tone, a header block and the program data itself. The header block and the program data are both repeated, but this redundancy is only used for the purposes of verification. If the two copies of program data do not match, a load error is generated and it is not possible to run the program. At the byte-level, an odd parity bit provides error detection (but not correction) capabilities.

The lead-in tone at the beginning of the recording of approximately 2840 Hz consists of repeated short pulses which allow for a correction factor to be calculated, permitting some variation in the speed of the magnetic tape (Hampshire, 1983).

Individual bits of data and the control signals to indicate the start of each byte and the end of the data (or header) block are encoded as pairs of short, medium or long duration pulses, as shown in Table 1.

Table 1. Meaning of pairs of pulses.

Meaning	First Pulse	Second Pulse
Lead-in	Short	Short
Start of a new byte	Long	Medium
0	Short	Medium
1	Medium	Short
End of header/program data	Long	Short

Table 2. System clock frequencies. (Parker, 2012, p. 27)

System	Video Mode	Frequency (Hz)
Commodore VIC-20	PAL	1,108,405
	NTSC	1,022,727
Commodore 64 and 128	PAL	985,248
	NTSC	1,022,727
Commodore 16 and Plus/4	PAL	886,724
	NTSC	894,886

TAP File Image

This image file format arose as a result of the work of Håkan Sundell in his Master's thesis (Sundell, 1996). The data in a TAP file represent the length of pulses on magnetic tape, expressed in terms of the system clock cycle frequency (divided by eight).

The data are encoded identically by the various Commodore machines, at a rate dependent upon the system clock. The system clock differs slightly between machines depending on whether the machine uses an NTSC or PAL display, as shown in Table 2.

For the initial version of the TAP format, each byte value of data in the file represents a single pulse. The duration of a pulse in seconds is given by the following formula:

$$pulse\ duration = \frac{byte\ value}{\frac{1}{8} \times frequency} \tag{1}$$

where $frequency$ is a value (in Hz) chosen from Table 2. For example, the greatest duration pulse which can be represented by a TAP image for the PAL VIC-20 platform is given by a value of FF (in hexadecimal). This corresponds to a duration of

$$\frac{FF_{16}}{\frac{1}{8} \times 1108405} = \frac{8 \times 255_{10}}{1108405} = 1840\mu s \tag{2}$$

which suffices to accurately capture the pulse data, except for the pauses between files. Since the gaps between files do not carry data, it is less essential for their duration to be precisely recorded.

Later versions of the TAP format included a means to accurately represent longer pulses. A zero value indicates the next three bytes in the file should be considered as the number of clock cycles corresponding to the duration of the pulse, in little-endian format. The distribution of TAP pulse values in image generated by wav2tap software is shown in Figure 2. From left to right, the three distinct peaks represent short, medium and long pulses, respectively. The principal advantage of the TAP format is that it is essentially a lossless format that preserves the cassette data as it occurs on the tape, instead of merely the bytes which make up the executable. This means that all aspects of the software are preserved, such as custom loading routines (e.g. ‘turbotape’), the music/graphics present whilst the program is loading and also any recording scheme violations (e.g. for implementing copy protection).

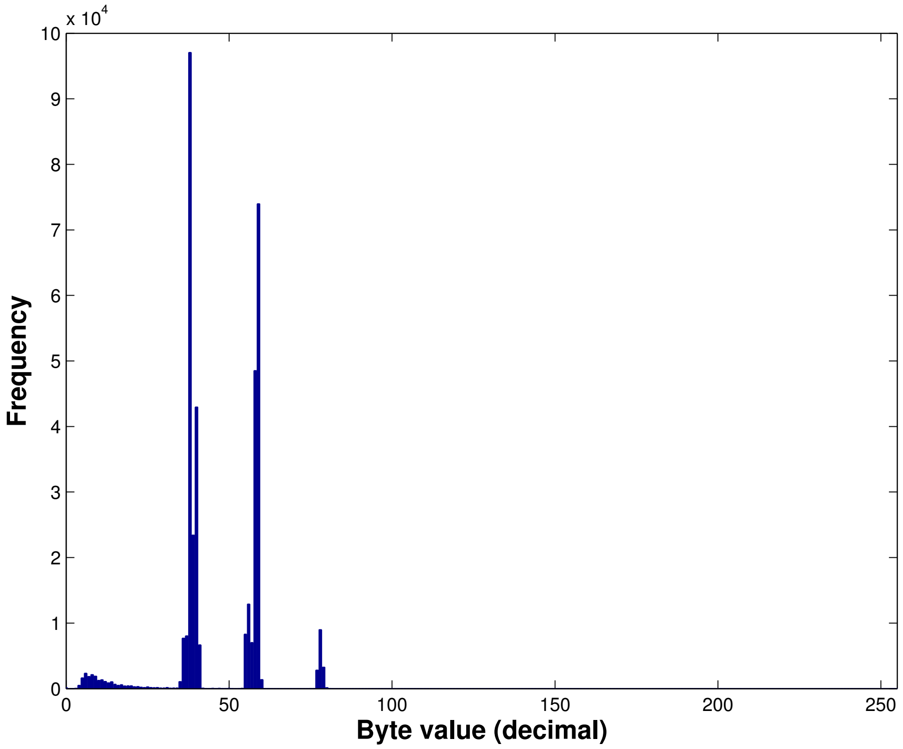

Figure 2. Distribution of TAP pulse values in image generated by wav2tap software.

T64 File Image

The T64 file format was developed by Miha Peternel (Peternel, 1991) as a part of the C64S emulator. It serves as a container for one (or more) Commodore programs (or data) encapsulating them in a single file on another system.

This representation is at a higher level of abstraction than TAP files, basically recording the values of the bytes of data directly rather than the pairs of waves which make up the individual bits.

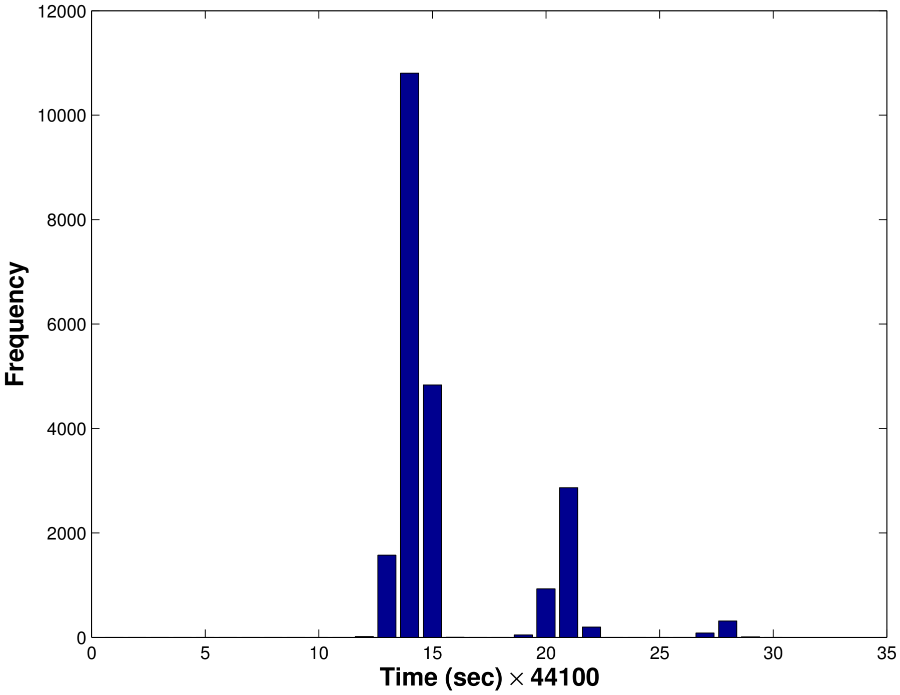

Figure 3. Histogram of pulse duration for MATLAB decoder (for a portion of the tape) where pulse duration is expressed as the number of samples between zero crossings.

Sampling Rate of Audio Recording

According to the Nyquist-Shannon sampling theorem, a recording of the audio cassette data at the lowest commonly used rate of 8000 Hz suffices to capture the underlying data, since the shortest wavelength pulse has frequency of approximately 2840 Hz. According to the author of the wav2tap program, a sample rate of 16 kHz or higher is recommended for the best results (Williams, 2010). In practice, a smoother waveform is obtained by sampling at the higher rate of 44.1 kHz, albeit at the cost of a slightly larger file size.

Methodology

Capturing Commodore Cassette Data as Audio

Initially, an inexpensive ‘Digitech Audio’ brand cassette player (model JW-51SWG) with a USB interface was used to make a recording of the cassette at 44100 Hz (bit depth of 16 bits). The resulting audio file in Waveform Audio File Format (WAV) was processed using WAV-PRG 4.0.1 (Gennari, 1998) to yield an image in T64 format. Using the VICE emulator (Boose et al., 1998) to run the program files on this image resulted in the game executing, but crashing a fraction of a second into gameplay.

The fact that the game executed at all suggests the vast majority of the data were correct, however some errors were present. It was decided to attempt to record the cassette again using better equipment. The State Library of South Australia provided access to a professional tape deck, a Tascam 122 MKIII connected to a Dell Precision T7500. The signal path was:

- Playback on Tascam 122 MKIII with stereo +4dbu balance analogue audio output,
- Analogue to digital conversion on Digital Audio Denmark ADDA 2402 high end professional 24bit 96kHz PCM Analog to Digital - Digital to Analogue converter with AES/EBU digital output to computer,
- Digital input to computer via digital AES/EBU transfer into RME HDSpe AES high end professional PCI bus audio interface card.

The process of digitising the Commodore data cassette was:

1. The cassette was fast wound back and forth a few times to reduce print through,
2. The azimuth alignment of tape head on Tascam 122 MKIII was adjusted to match record azimuth as closely as possible,
3. The tape playback pitch adjust set to off position (i.e. standard playback speed of $1\frac{7}{8}$ i.p.s.),
4. Levels into ADDA set to maximum possible level without digital overload,
5. Software (Steinberg Wavelab[1]) set to 24 bit depth, 96 kHz sample rate and file type uncompressed .wav,
6. Recording made of full data track in stereo.

This was then downsampled to 44.1 kHz, using Steinberg's Wavelab, for compatibility with WAV-PRG.

Custom Decoder

MATLAB was used to find the sampled point just before the falling edge zero crossing. That is, the sample point just above the horizontal axis. Though interpolation between samples could have been used to find a more precise estimate of where the pulse actually begins by estimating the intersection between the waveform and the horizontal axis.

Using the point just before the true beginning of the pulse proved sufficiently accurate to classify a pulse, given a sample rate of 44100 Hz. A histogram was generated to show the distribution of pulse lengths and manually estimate the cutoff points used to classify a pulse as either short, medium or long. For example, the duration of the pulse shown in Figure 1 as indicated by the dashed line corresponds to a duration of $\frac{28}{44100}$ seconds, or approximately 635 microseconds, which was recorded as a value of 28 in the histogram shown in Figure 3.

Before creating an image in TAP format, the MATLAB script performed the additional step of decoding the data to the byte level. The parity check bit for each byte was used to verify the data were correct.

[1] Steinberg Media Technologies: http://www.steinberg.net/en/products/wavelab/start.html

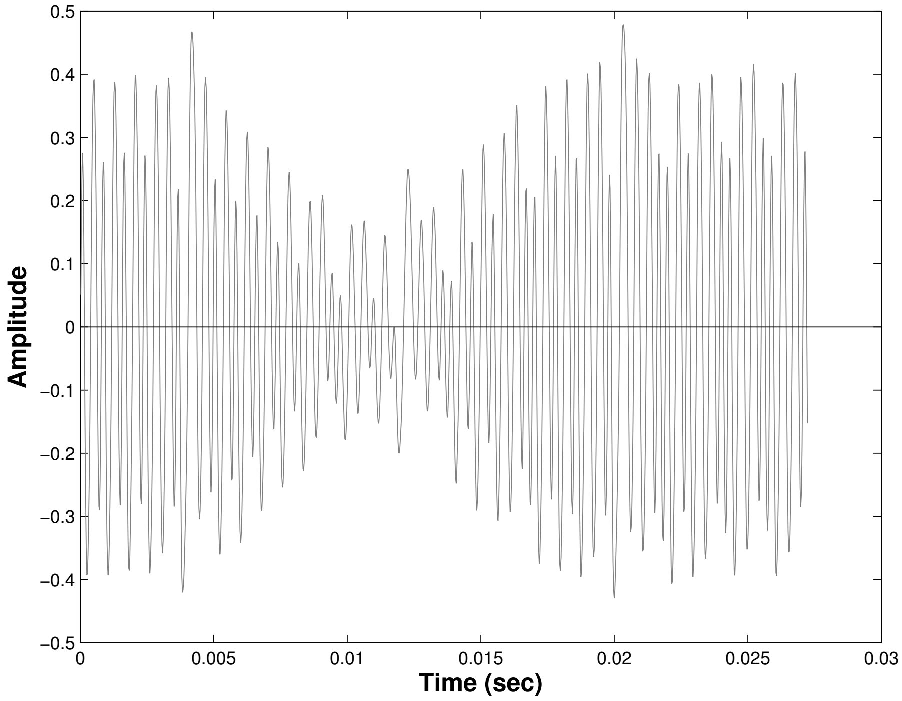

Figure 4. Plot of tape waveform sampled at 44100 Hz illustrating dropout.

Results

The tape image produced in T64 format by using WAV-PRG was playable. However, once the cassette is in this form it becomes more difficult to correct errors since only the executable code is stored in the image. The redundant copies of the header and program data are discarded.

An image in TAP file format was created using wav2tap (Williams, 2010) which contained a single uncorrected error in a parity check bit. This error was a result of a dropout in the audio, which is briefly described in the following section.

Dropout

The phenomena of dropout when reading magnetic tape produces a characteristic change in the shape of the waveform as if it has been pinched together, as shown in Figure 4. Sarigoz, Kumar and Bain 2001 echo the findings of Hoagland et al. (Hoagland, Oehme & Talke, 1978) and Osaki et al. (Osaki, Kurihara & Kanou, 1994) in stating that the principal causes of dropouts has been found to be loose particles present between the media and the tape head or defects in the media itself. Simply put, a small bubble (or bulge) causes the distance between the tape and the head to gradually increase, resulting in a drop in the level of signal.

Whilst dropouts may cause long trains of bit errors, known as ‘bursts’ (Sarigoz, Kumar & Bain, 2001), in this instance, the short interruption of less than 0.02 seconds duration was sufficient to cause only a single bit error. The peak level of distortion in the shape of the waveform is at the middle of the dropout caused a portion of the signal to be pushed below the axis, resulting in the disappearance of a zero crossing. This anomaly

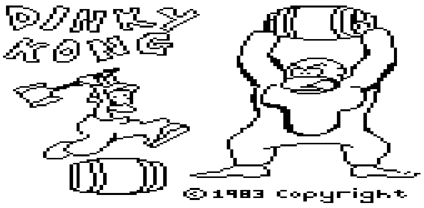

Figure 5. Title screen displaying in VICE emulator.

was detected in the MATLAB script as an outlier in the pulse width data. Basically, two adjacent pulses were merged into one pulse which had approximately double the duration of any other pulse.

Sarigoz, Kumar and Bain 2000 show the peakshift profile may be used to detect dropout since we expect the peaks in the waveform to remain reasonably constant over time. Peakshift is defined as "the amount of the time a pulse in the readout signal is shifted away from its nominal location by the dropout event" (Sarigoz, Kumar & Bain, 2000).

Sarigoz et al. 2001 provides an exponential model linking variation in the shape of the waveform, in terms of its amplitude and peak location to the variation in spacing between the head and tape. This model is used to implement a dropout correction scheme which can detect and correct the dropout in real-time.

In this instance, since the error occurred in the odd parity check bit itself, rather than the data, the missing bit was able to be recovered under the assumption that the previous eight bits were correct. No other discrepancies were found, meaning there was a single error over the entire recording. The error was of no consequence because it was located in the second (redundant) copy of the program data. The resultant file image was executable in a VICE emulator, the title screen image of which is seen in Figure 5.

Conclusions

In this article, knowledge of the Commodore tape data format to was used to decode, verify and correct errors on a magnetic tape recording. The output produced by MATLAB and Perl based scripts was an error-free TAP file image compatible with the VICE emulator.

The initial recording, made on lower quality equipment, contained many more dropouts, including one in the leader tone which was detectable by ear as a change in pitch. However, there was none in the exact spot corresponding to the single dropout in the final recording. We suspect the single error encountered in the final recording was more than likely a result of debris between the playback heads and the tape itself. This suggests that more than one recording may be necessary, along with some compositing of audio, in order to obtain a file from which the original data may be recovered.

Acknowledgements

Craig Harrington died 7 October 2014.

'Play It Again: Creating a Playable History of Australasian Digital Games, for Industry, Community and Research Purposes' is supported under the Australian Research Council's Linkage Projects funding scheme (project number LP120100218).

The authors thank Peter Kolomitsev, from Audio Preservation, State Library of South Australia, for his invaluable assistance.

References

Boose, A., Biczo, T., Lem, D., Dehmel, A., Matthies, A., Pottendorfer, M., ... Sonninen, J. (1998). VICE: The Versatile Commodore Emulator. Retrieved from http://vice-emu.sourceforge.net

Commodore Business Machines Inc. (1984). *Service manual datasette model C2N/1530/1531*.

Fraia, L. D. (2006). DC2N - A tape preservation device for Commodore and Sinclair homecomputers. Retrieved from http://www.luigidifraia.com/c64/dc2n/

Gennari, F. (1998). WAV-PRG and Audiotap. Retrieved from http://wav-prg.sourceforge.net

Georgiou, V. (1984). *Commodore 64 interfacing blue book*. Microsignal Press.

Gibson, G. D. (1997). Magnetic tape deterioration: recognition, recovery and prevention. *United Nations Educational Scientific and Cultural Organisation Publications - CII WS*, 259–271.

Guttenbrunner, M., Ghete, M., John, A., Lederer, C. & Rauber, A. (2009). Digital archeology: recovering digital objects from audio waveforms. In *Proceedings of the sixth international conference on preservation of digital objects (iPRES 2009)*.

Hampshire, N. (1983). *VIC Revealed*. Hayden Book Co.

Hoagland, A., Oehme, W. & Talke, F. (1978). Narrow track defect studies on flexible media. *IEEE Transactions on Magnetics*, *14*(5), 740–742.

Osaki, H., Kurihara, J. & Kanou, T. (1994). Mechanisms of head-clogging by particulate magnetic tapes in helical scan video tape recorders. *IEEE Transactions on Magnetics*, *30*(4), 1491–1498.

Parker, N. (2012). Free Commodore computer magazine. Retrieved from www.commodorefree.com/magazine/vol6/issue59.pdf

Peternel, M. (1991). C64s Commodore 64 emulator for PC. Retrieved from http://www.zimmers.net/anonftp/pub/cbm/crossplatform/emulators/msdos/c64s/c64s.readme

Sarigoz, F., Kumar, B. & Bain, J. (2000). Characterization and equalization of dropouts in the magnetic tape recording channel. In *Acoustics, speech, and signal processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference* (Vol. 6, pp. 3554–3557). IEEE. doi:10.1109/ICASSP.2000.860169

Sarigoz, F., Kumar, B. & Bain, J. (2001). Performance of dropout correction on real magnetic tape waveforms with dropouts. *IEEE Transactions on Magnetics*, *37*(2), 639–645. doi:10.1109/20.917594

Sibley, M. (2014). Monkey X. Retrieved from http://www.monkey-x.com/

Sundell, H. (1996). *Correct emulation of micro-computer C64 in software independent of computer system* (Master's thesis, University College of Boros).

Williams, C. (2010). C64 Datasette tape utilities. Retrieved from http://sourceforge.net/projects/c64tapedecode/