

Image Glitching: Manipulating Images with Audio Effects

Erik Brunvand
University of Utah
School of Computing
elb@cs.utah.edu

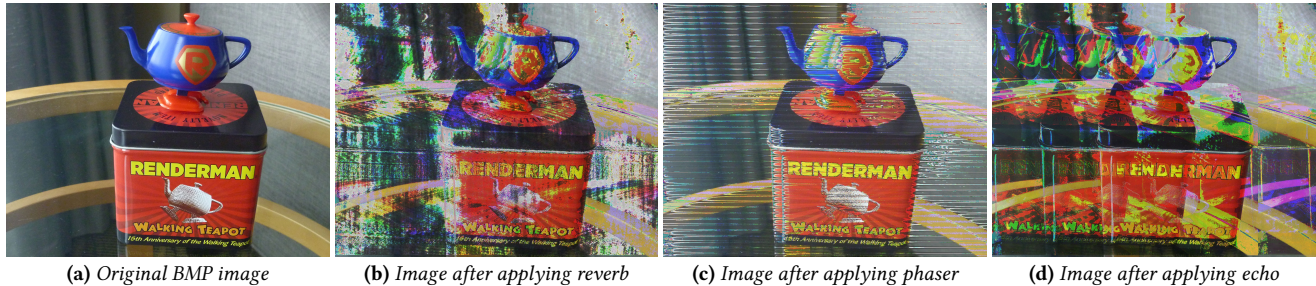


Figure 1: Examples of applying different audio filters on image data. Images are imported into the audio editor Audacity [Audacity 2018] as raw data, manipulated as if they were audio files, then exported back to images to see the visual effects of the audio filters.

ABSTRACT

Image glitching and data-bending are used to introduce image formats, data manipulation, and data visualization to beginning CS students and non-major students taking computing courses with no coding required.

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; • **Applied computing** → *Fine arts; Sound and music computing*.

KEYWORDS

Image glitching

ACM Reference Format:

Erik Brunvand. 2019. Image Glitching: Manipulating Images with Audio Effects. In *Proceedings of SIGGRAPH '19 Educator's Forum*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3326542.3328017>

1 OVERVIEW

This Groovy Graphics Assignment lies at the intersection of computer graphics, media computation, and fine art. It uses image glitching [Wikipedia contributors 2018b] and data-bending [Wikipedia contributors 2018a] to introduce image manipulation and data representations to students ranging from non-majors learning computing, to fine art majors, to computer graphics students. Data bending is a way of thinking about data that allows it to be manipulated, or bent, in ways not normally associated with that form of data, usually for aesthetic rather than scientific purposes. Often this

takes the form of using a tool designed for one type of data (audio data, for example) and using it to manipulate another form of data (e.g. images). Image glitching is a form of artistic expression that manipulates images through intentional or unintentional glitches - mistakes or random modifications to the data. In recent years image glitching and other forms of glitch-art have captured the imagination of a variety of artists and art critics (e.g. [Geere 2010; Menkman 2011; Nunes 2011]).

From an educational perspective, image glitching and data bending are powerful tools that can be used in courses for non-majors (e.g. [Brunvand and McCurdy 2017]), in media computation courses for computer science majors (e.g. [Guzdial 2003; Simon et al. 2010]), and in fine art courses that have a technological bent (e.g. [Brunvand and Stout 2011]). The process of data bending images with audio filters illustrates many principles about data representation and data manipulation, and does so with no programming required. The assignment may be completed using all open-source tools.

2 IMAGE GLITCHING PROCEDURE

There are many ways to apply a glitching aesthetic to images. Image data may be manipulated directly using programs or even using text editors. Images may also be degraded through purposeful use of lossy compression techniques. The technique in this assignment is to use data bending ideas to manipulate image data in an audio editor, then export that modified data back to an image file. To do this the image first needs to be converted to an uncompressed format such as .bmp, .raw, or .tif. Compressed image formats such as .jpg may be bent in other ways, but the process of applying audio filters will likely make the .jpg unreadable. Conversion of an image to an uncompressed format is easily accomplished in a tool such as Photoshop or an open-source image tool such as Gimp.

Once the image has been converted, it can be imported to an audio editing tool such the open-source Audacity [Audacity 2018]. In Audacity, students use File→Import→RawData to import the image into the audio editor. There are many encoding choices and

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGGRAPH '19 Educator's Forum, July 28 - August 01, 2019, Los Angeles, CA, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6782-0/19/07.

<https://doi.org/10.1145/3326542.3328017>

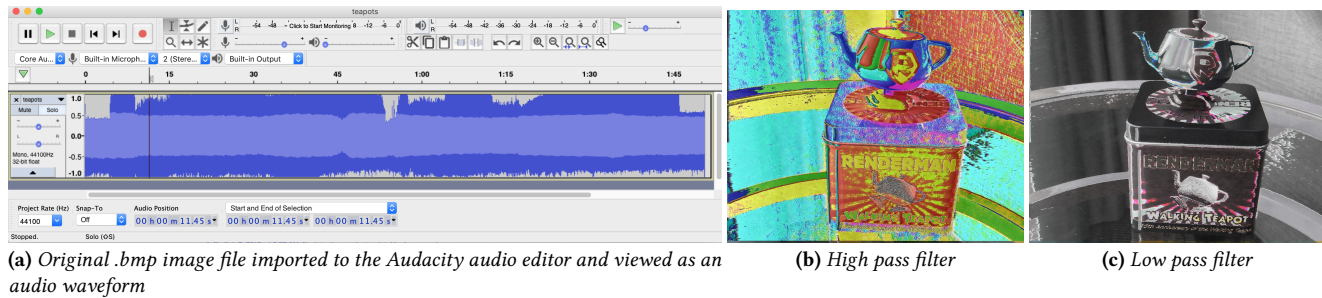


Figure 2: Audacity interface (a) and the result of applying frequency filtering to the image (b) and (c).

the most important thing is that the same format is used for import and export. We have had good luck with U-Law (μ -Law) encoding (which preserves much of the dynamic range of the image) and Big-Endian format. Figure 2(a) shows the Audacity interface with the “walking teapot” image from Figure 1(a) imported as audio data. Interestingly, this image may now be “played” as an audio file - some images make much more interesting sounds than others!

Once imported into the audio editor, audio effects may be applied to the data using the Effects menu. These filters use digital signal processing (DSP) algorithms to modify how the audio sounds. At this point it is critical that the first quarter second or so of the data *not* be modified. This is the file header and if effects are applied there the exported file will no longer be recognized as an image file. Thus, the cursor is used to select all but a small portion on the far left of the “sound” waveform. The audio effect will then be applied only to the image data, not the file header.

Once the effect, or multiple effects, have been applied, the data may be exported and the results of the audio filters seen in the image. To export select the File→Export→ExportAudio option and make sure to select RAW (Headerless) formatting, and the same encoding as used in the import (e.g. U-Law). This data will be exported as a .raw file. I find that it’s easiest to edit the file name back to .bmp as a format for easiest viewing of the image. Figure 1 shows the result of this procedure. The leftmost subfigure (a) is the original image in .bmp format. The other three subfigures have been manipulated with a single audio filter: reverb, phaser, and echo respectively. Figure 2 (b) and (c) show the result of using audio high-pass and low-pass filters on the image data. Of course, multiple filters may also be applied to the image for even more drastic image modification/glitching results.

3 DISCUSSION

This image glitching assignment requires only that the students are able to run open-source tools on their machines such as Gimp and Audacity (other tools such as those in the Adobe suite will also work). The assignment provides opportunities to talk about a wide variety of issues related to image data, data formatting, data manipulation, signalling issues such as frequency and bandwidth, and even topics relating to the conjunction of arts and technology. We use this in non-major courses to talk about image formats (RGB pixel encodings, compressed vs. uncompressed formats, lossy vs. lossless compression etc.) and also about signal processing topics such as audio data encoding and audio effects processing. In com-

Table 1: Metadata in tabular format

Summary	Image glitching through data bending
Learning Outcomes	Image data formatting and manipulation without coding. Glitch-based image manipulation.
Classification	Fundamentals, image processing, visualization
Audience	CS0, CS1, Computer Graphics, Fine Arts
Dependencies	Basic use of image/audio applications
Prerequisites	None
Strengths	Eye opening to data-bending possibilities. No coding required, and visually extremely interesting results are generated.
Weaknesses	Care must be taken to not corrupt image files
Variants	Many variations on combinations of audio filters, and extensions to more advanced data manipulation
Assessment	May be assessed as an aesthetic image, or as a technical visualization of the filter

puter graphics courses we use this to encourage experimentation and as a precursor to writing application programs that directly generate and manipulate image files. In fine arts courses this is an opportunity to introduce data-based and glitch-based new-media art to both art students and general ed students, and to add a technological literacy component to that discussion through exploring the data without having to write code.

REFERENCES

- Audacity. 2018. Free, open source, cross-platform audio software. <https://www.audacityteam.org/>. (2018). [Online; accessed 11-Feb-2019].
- E. Brunvand and N. McCurdy. 2017. Making Noise: Using Sound-Art to explore Technological Fluency. In *ACM SIGCSE'17*. Seattle, WA USA.
- E. Brunvand and P. Stout. 2011. Kinetic Art and Embedded Systems: A Natural Collaboration. In *ACM SIGCSE'11*. Dallas, TX USA.
- Duncan Geere. 2010. Glitch art created by 'databending'. <https://www.wired.co.uk/article/glitch-art-databending>. (2010). [Online; accessed 10-Feb-2019].
- Mark Guzdial. 2003. A Media Computation Course for Non-majors. *SIGCSE Bull.* 35, 3 (June 2003).
- Rosa Menkman. 2011. *The Glitch Moment(um)*. Inst. of Network Cultures, Amsterdam.
- Mark Nunes (Ed.). 2011. *Error: Glitch, Noise, and Jam in New Media Cultures*. The Continuum International Publishing Group, New York.
- Beth Simon, Päivi Kinnunen, Leo Porter, and Dov Zakzakis. 2010. Experience Report: CS1 for Majors with Media Computation. In *ITICSE '10*.
- Wikipedia contributors. 2018a. Databending — Wikipedia, The Free Encyclopedia. <https://en.wikipedia.org/wiki/Databending>. (2018). [Online; accessed 10-Feb-2019].
- Wikipedia contributors. 2018b. Glitch art — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Circuit_bending. (2018). [Online; accessed 10-Feb-2019].