Precise Mishandling of the Digital Image Structure

Theodore Davis

The Basel School of Design, HGK FHNW,
Vogelsangstrasse 15, 4058 Basel, Switzerland
ted@teddavis.org

Abstract. In the everyday capturing and sharing of digital images, one is rarely confronted with the notion of this media being any different than its analog counterpart. With the exception of an occasional compressed artifact, the medium remains transparent to the pictorial content it contains—that is until an error or glitch occurs, providing a brief glimpse into the inherent properties of the medium itself. However, through advancements in error detection techniques, these artifacts are seen with less and less frequency, thus further removing the message from its medium. What if the glitch was caused intentionally? What range of innovative visual possibilities might unfold as the digital image is pushed beyond its ideal state? In pursuing these questions, the term precise mishandling was coined to describe a method of carefully and thoroughly manipulating the structural code found within a digital image, meanwhile observing the reaction this had on the pictorial surface. Utilizing the most common file format for digital images, the JPEG, this paper follows the discovery of a technique for generating innovative imagery through the cross-media visualization of text.

Keywords: new media, media studies, interface design, design process, image design, digital image, jpeg file format, precise mishandling, glitch, cross-media visualization, text2image.

1 Introduction

Through the use of commercial design software, most of which offer tools analogous to hand techniques, the range of visual possibilities is, to no surprise, nearly identical to that of their analog inspirations. Further exemplified by the reoccurring reference to a *canvas*, this selective focus onto the surface of an image greatly ignores the digital code of which the medium is entirely composed. Similar to the function of a window, through its transparency, it allows one to view the outside surroundings and is rarely noticed until it becomes excessively dirty or a crack occurs. At this point, the window transfers from a transparent or unnoticed medium to an opaque one. As common file formats such as the JPEG/GIF/PNG are used to store and share digital images across the Internet, awareness of the media itself is rare, albeit for minor artifacts of compression that may occur through each exchange. To the average user, this process is seemingly mundane—that is until a glitch occurs along the path of transmission, in

A. Marcus (Ed.): Design, User Experience, and Usability, Pt I, HCII 2011, LNCS 6769, pp. 211–220, 2011.

other words generating a crack in the window. As the content is disturbed, this rarely desired error produces an unpredictable visual, potentially revealing inherent qualities within the file format itself.

Through the advances of error detection and correction techniques, such a phenomenon occurs with less and less frequency, thus further removing the message from its medium [1]. However, what if the glitch was caused intentionally, manipulating the data file directly, thus circumventing the system set in place to counterbalance it? What range of visual possibilities might unfold as the medium is pushed beyond its *ideal* state?

In the pursuit of answering these questions, the term *precise mishandling* was coined in order to fit the proposed methodology. That is, the intentional and systematic manipulation of a stable system, in order to learn how it functions and to discover potential exploits for creative use. It was through this technique that an investigation of the digital image's *underlying code* [2] began, soon leading to the development of tools capable of generating a wide range of new visual possibilities.

2 Precise Mishandling in Recent Years

The term, *precise mishandling*, can be useful in describing artistic practices in which the error or *glitch* is caused with intention for creative expression. Within the fields of music [3] and visual arts [4], the glitch offers unfettered access to the unexpected and uncontrolled within a computerized or electronic system. While occurring naturally and with great spontaneity, artists have harnessed and utilized the glitch in what could be seen as a rejection of the pixel perfection that exists in our digital representation of analog artifacts. This could be found in recent years through a popularized experimental technique dubbed by many as *datamoshing* [5] in which the key frames of a movie file are removed, forcing the computer to blend pixels at will, everytime a dynamic change in the moving action takes place.

A precedent for specific file format manipulation can also be found in Cory Arcangel's 2003 work *Data Diaries* [6]. Created in a time of minimal error correction, the work consisted of removing the entire data contents from a *Quicktime* video file format, while maintaining the header used for playback instructions. According to his explanation [7], once the original content was removed, the player defaulted to displaying the computer's daily stored RAM as though it were video data. The result was a streaming array of pixels, waves and bands, as 128 megabytes of random access memory were spontaneously transcribed. This repurposing of media could be considered a form of *cross-media visualization*, in which, one form of digital media merely acts as a vessel for the display of another.

3 Exploring the Digital Image Medium

When considering the many formats a digital image can embody, the JPEG or *.jpg*, is without a doubt the most common in use. Its prevailing application can be found as the default file type used by the majority of digital cameras as well as when sharing images across the Internet. This is likely due to the standards having been set at an early stage of digital adoption, along with a compression scheme that provides a significant reduction in file size while maintaining photographic detail. As the file type remains virtually the same since its inception in the early nineties, it became the perfect candidate for exploration. While, in the beginning of this research, a series of tests were conducted on the wide range of digital image file formats currently in use,[1] the JPEG proved to be the most susceptible to manipulation, which is likely due to an entropic encoding and decoding process.

{digital image} » {*encoder specifications*} » {compressed jpeg file} .
{compressed jpeg file} » {*decoder specifications*} » {visible image} .[2]

3.1 Surface versus Structure Relationship

A core attribute that separates the digital image from its analog counterpart is that it could be devised into two modes of representation, the image surface and structural code. The image surface refers to the pictorial quality that is rendered by image viewing software, allowing it to "participate in dialog with other cultural objects" [8]. Whereas the structural code consists of the rudimentary ones and zeros of binary code, in which nearly every digital file is composed of [9]. Before this data is displayed as a digital image, it could be said to 'participate in dialog with other *computational* objects'. In other words, the binary representation of an image file is no different than that of a text, movie or sound file.

When it comes to comprehension, the computer remains limited by its near solitary understanding of the numerical code.[3] Humans on the other hand, have the ability to interpret both, even though out of convention only the surface level is read. With the

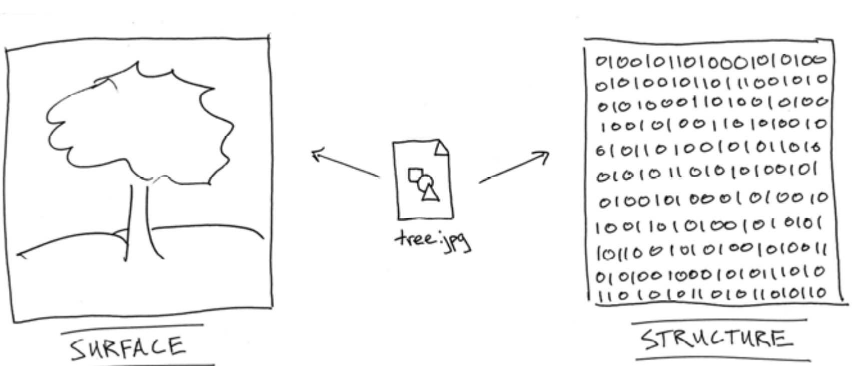


Fig. 1. The digital image file as represented by both its *surface* and *structure (underlying code)*

[1] JPEG, GIF, and PNG being the most commonly used for transferring bitmap images, while additional format tests included BMP, PICT, PDF, PSD, SGI, TIFF, and TGA.

[2] Based on *JPEG Standard recommended guidelines* of www.w3.org.

[3] While significant advances in Image Processing exist, getting beyond the lower-level attributes (color, texture, form), to a comprehensive contextual understanding by the computer, remains a distant challenge.

aid of software such as the hex-editor[4], a virtually unknown tool in the realm of visual communication, the aptitude to comprehend this structural layer is enhanced by the capacity to precisely modify the now visible code (see Fig. 1).

3.2 Comprehending the Code

Making the structural code of a digital image visible could be analogous to opening the hood on an automobile, particularly when one begins to poke and prod at the foreign matter found. Without prior experience, one is likely to break the image structure, thus rendering the file unreadable, or in the case of the automobile, immobile. However this analogy quickly dissolves as the computer, with its lack of an original in terms of duplication, can be used to learn through trial and error, while always maintaining a backup file to revert to. It was through this process that the technique of *precise mishandling* was developed. In this case, selecting every offset within the structural code and pushing its value from an initially stable state through the entire hexadecimal range (00 » FF or 256 values).[5]

Originally generating every hexadecimal variation by hand, followed by the use of a keyboard macro, and soon after a programmed script, a genetic schema of the digital image was thoroughly unraveled, resulting in a visual map of structural deviations. Parallel to reading technical descriptions about the JPEG file format [10], [11], [12], these illustrative examples of each described region now offered a better understanding from an outsider's point of view.

3.3 Dividing the Code

As is the case with all digital image files, the JPEG file format structure is divided into two distinct parts, a header and body (see Fig. 2). The *header*, which resides at

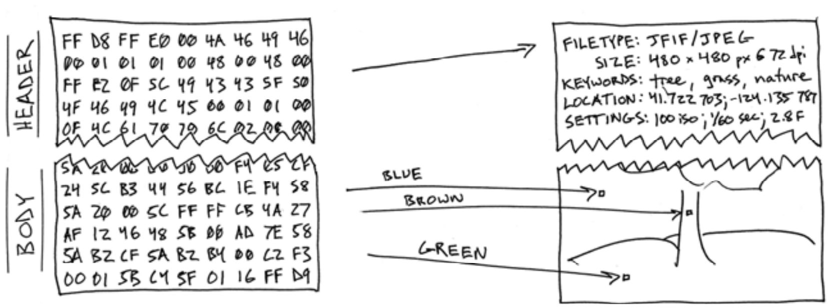


Fig. 2. The structural code is split into a *header* and a *body*. The *header* consists of rendering instructions along with metadata including declarations such as the file format used, dimensional size and resolution. The *body* is composed of literal clauses addressing specific pixels and their visual attributes.

[4] A hex-editor allows the user to view and modify the binary code (represented as both hexadecimal and ASCII character format for ease) of any data file. This software is often used in the field of digital forensics.

[5] Visit *www.ffd8.org/header_x256* to see the entire range of visual results for one particular region of JPEG file format, the *quantization table*.

the top of the image, can be described as instructional information read by any given software to decipher the type of data being processed and how it should be rendered. The *body*, which follows a designated marker signifying the completion of the header, is responsible for the actual content being processed or in this case, the actual picture. While research was conducted to both regions in parallel, this paper focuses on the findings within the *body* region.

4 Precisely Mishandling the Digital Image Body

Through learning to decipher the various regions of the JPEG structure, it became clear that the primary textual information found within the digital image, *metadata*, was being stored within the *header*, therefore sitting on the periphery of the actual pictorial contents. While this is completely logical considering the information must pass, untouched, through a decoding/encoding process, it leaves opens the potential for precisely mishandling the placement of such text. Specifically, to explore the cause and effect relationship that might occur when injecting textual content directly into the *body* of the image, prior to the decoding process. Would it be possible for the text to find its way, in recognizable form, to the image surface—or might the JPEG simply reject the content and refuse to render?

Based on the decoding process a JPEG file goes through, the hypothesis was formed, that if the *header* or instructions for rendering the image were kept intact, the system would have no choice but to display injected information in the exact same way as it would with existing data. This thesis proved correct immediately following the discovery of where the header ended and the body began.

4.1 Con*text*ual Images

With the body of the digital image now isolated, textual information could be inserted at any point within this code. Initial experiments involved the pairing of texts that had a contextual relationship to the photographs they would inhibit. This was first done with the use of presidential portraits and an accompanying transcript from the inaugural speeches given at the beginning of each term in office (see Fig. 3). Utilizing a raw unformatted text of each speech, a subjective decision was made as to where in the image this text would begin. An offset corresponding to the pictorial surface location of their mouth was used for added association. Since each Ex-President had held two terms of office, this offered two sets of sample data to inject and compare.

C_0 C_1 C_2 B_0 B_1 B_2

Fig. 3. Transcripts of the now Ex-Presidential inaugural addresses were injected into their portraits (*C_0 and B_0*) as found on whitehouse.gov in 2008. The first and second term speeches can be found in the corresponding numbered suffix, *C_1 and C_2*, respectively.

Upon the implementation of this experiment, one particular variation that occurred in both cases was the brightness of tonal value decreasing as the picture and text flow towards the bottom. As expected the hue initially begins with the same value as the mouth region in which it began, however with the first speech, this value changes both in brightness and adopts color values that were never present in the original image. This trend does not occur in the second speech, in which the tonality remains nearly the same from the mouth until the bottom of the image.

Expanding upon this study, similar experiments were conducted using an official photograph of The White House in combination with the 37 inaugural speeches given by former US Presidents. The symbolic offset of 1600 was selected[6] and each inaugural speech was injected into the same location within the image structure. Additional studies were created through the convergence of MySpace profile photos and their profile descriptions. This was done with the ambition of visualizing the parallel or disconnect in ones projected appearance both in terms of pictorial and textual quality.

With each additional experiment, so grew the impression that a contextual visualization of the text was being developed. This gave the hope that one could extract a deeper meaning from the images being produced. However, this conviction became less certain as the location of injected text was altered, even with the slightest change (see Fig. 4). In the White House experiment, changing the offset location of injected text from an initial 1600 to 6400 or further to 12800 changed the resulting image dramatically.

J_0 J_{1600} J_{6400} J_{12800}

Fig. 4. From an initial web resolution photograph of The White House (J_0) a transcript of Thomas Jefferson's first inaugural speech was injected into the digital image structure at offset 1600 (J_{1600}). This was subsequently followed by injecting the original digital image at offsets 6400 (J_{6400}) and 12800 (J_{12800}), respectively.

It was at this point that the decision to work with smaller lengths of text arose, thus isolating the visual deviation of each minute change. Individual characters were soon used to replace the entire pictorial contents within the code and thus the medium truly became the massage.

4.2 TEXT2IMAGE

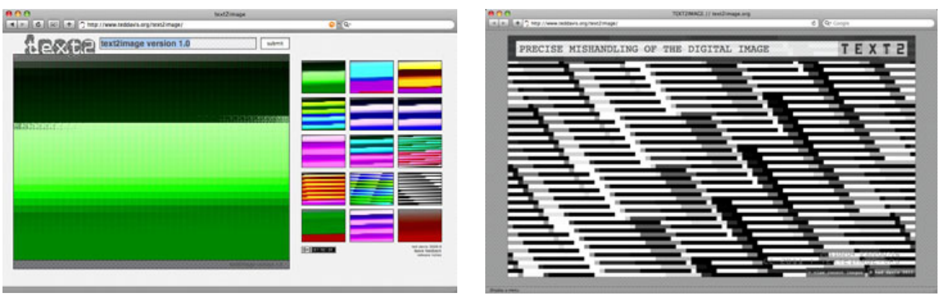
While the initial process of injecting text into the digital image was done manually, it took an absorbent amount of time to generate each new variation. Along the process of *precise mishandling*, a state of *exploration* shifted to *discovery*, at which point it became time to find an automated way to accomplish the task.

[6] 1600 Pennsylvania Ave. NW, Washington, D.C., is the physical address of The White House.

Motivation for developing an automated tool also stemmed from an interest in exploring and cataloging imagery that utilized the widest gamut of textual input possible. This included input from other languages, keyboard characters beyond the norm, and all content that would have never been conceived by one person alone.

Implemented as a web application using a LAMP (Linux, Apache, mySQL, PHP) setup, *TEXT2IMAGE* [13], a globally accessible and platform-free tool, was born. To great surprise, within the first five days of being launched, it amassed more than 13,000 images—20,000 by the second week. Designed with simplicity and universal access in mind, the interface consists of little more than an empty input textbox and a large colorful image filling the page (see Fig. 5). Upon loading the website, the textbox is automatically focused upon, ready to accept virtually any textual input from the user's keyboard by utilizing the UTF-8 character set. The enter key or submit button is pressed and within seconds the main image refreshes, displaying a new image, generated entirely from the provided text. Modifying the text ever so slightly has the potential to shift the image in dramatic ways, however the rendered image is always consistent for a given text.

Extended features of *TEXT2IMAGE* include a *recent images* tab, showcasing an array of the last fifteen generated images. Hovering the mouse over each thumbnail displays the text that was used to create it, whereas clicking will replace both the main image and textbox with the selected output. This allows one to continue exploring from a point of existing visual inspiration. Lastly by clicking on the main image, the tool cycles to a second algorithm, which produces an alternative, black and white, visualization in parallel.

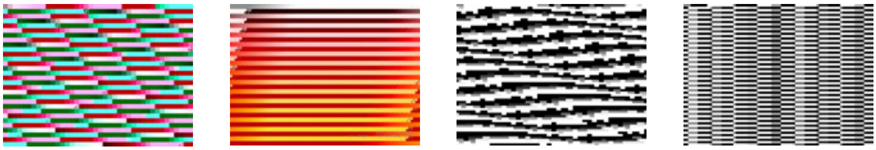


V_1 V_3

Fig. 5. This shows the progression of the *TEXT2IMAGE* interface from its initial release (V_1) to its latest refinement in version 3.0 (V_3)

4.3 Evaluating the New in Precisely Mishandled Images

Aspects of the digital images created through *TEXT2IMAGE* tend to be reminiscent of the 8-bit graphics found in video game consoles from the early 1980's. Where this deviates is in the complex graduating hues and progressive repeating structures that compose these block-like pixels (see Fig. 6). Between the two algorithms available within the tool, the first one returns a greater surprise in the variety of immediate visual output through the wide variations in color values. The output of the second algorithm is seemingly less varied, albeit it contains a more complex visual structure. Optical illusions of three-dimensional depth are often experienced through an adjustment of scale and distance from the image.



alg_01$_1$ *alg_01*$_2$ *alg_02*$_1$ *alg_02*$_2$

Fig. 6. A sample view of output images from both the color algorithm (*alg_01*) and the black and white algorithm (*alg_02*)

Throughout the research phase leading up to *TEXT2IMAGE*, a frequently asked question was how the newly generated images could be graphically applied. As the website began to be covered in blogs and traffic surged, the concern was drowned out by a general attraction in both the technique of creation and the resulting imagery. This notion was well observed as one particular blog wrote, "*text2image is one of those 'data art' works that balances on the line of total uselessness and visual fascination. While there is no immediate purpose, one still spends minutes playing around with it.*" [14]. In contrast it was also described as a medium in its own right for artistic expression, "*Watercolor, oil paints, charcoal, sanguine, India ink. . . There exists a large palette of techniques for those who want to unleash their creativity and create an image. Ted Davis adds a new one: the Web!*" [15].

Hypothetical applications for the imagery were continually pursued, including re-upholstered Ikea furniture catalogs, personalized apparel and revised visualizations for cookbook recipes. Nevertheless it was a half-year later, as a design student from the UK used *TEXT2IMAGE* to generate book covers for a design contest entry [16], that the produced imagery was once again dispersed across a new range of blogs. Migrating now from visualization and technology based blogs to those in the realm of graphic design, a confirmation grew in that the generated imagery contained previously unforeseen qualities.

5 Conclusion

Under the premise that nearly all of our digital files are speaking the same language of binary code, it became plausible to mix such contrasting phenomena as text and image together. In an analog domain, this could be done through the use of collage, physically layering one over the other. However in the digital domain, these elements have the capacity of being combined underneath the surface, on the structural level. By mixing the contents of a *cross-media visualization* was created. Through the *precise mishandling* of each file format, its requirement to act as a transparent medium is disregarded, as the inherent visual qualities it wishes to share are further explored and exposed.

As new visual discoveries were made through the introduction of forced glitches, a lack of proper tools to accommodate this practice was quickly realized. Ten years ago at the time of Kim Cascone's seminal text on the use of failure and glitches in electronic music [17], a variety of tools for the emulation and implementation of audible glitches already existed. A call was made in the concluding paragraph for the development of tools that would help to educate the musician in this field of practice, specifically beyond a "blind experimentation". Following on the heels of *glitch music*, *glitch art* [18], as it has grown to be known, severely lacks the tools for both exploration and education in usage of the glitch. While the hex-editor continues to be a usefully appropriated tool in order to accomplish both of these tasks, it is in dire need of specialized functions relative to the file types that visual media embodies. This inspired the development of *HEADer_REMIX* [19], an alternative hex-editor, limiting access to the safely malleable regions within the image header, as well as providing an instant visual update from the resulting changes to the pictorial surface. What it lacks in novice approachability, it makes up for through education in regard to the cause and effect relationship of the header and image surface. Perhaps one day, professional software will offer a sort of *digital native* user access, allowing for the selective disabling of error correction techniques. As glitches and bugs find their way into our working documents, an awareness for the media will grow, parallel to the development of a new visual language—as discovered through *TEXT2IMAGE*—accompanying our journey further into the realm of digitalization.

References

1. The Playboy Interview: Marshall McLuhan (1969), `http://www.nextnature.net/?p=1025`
2. Manovich, L.: The Language of New Media, p. 289. MIT Press, Cambridge (2001)
3. Cascone, K.: The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music. Computer Music Journal 24(4), 12–18 (2000)
4. Goriunova, O., Shulgin, A.: Glitch. Neural 28, 110–119 (2007)
5. Rhizome blog discussion, `http://rhizome.org/editorial/2009/feb/25/pixel-bleed/`
6. DATA DIARIEZZZZZZ, `http://www.turbulence.org/Works/arcangel/`
7. Columbia University Lecture, `http://www.columbia.edu/itc/soa/dmc/cory_arcangel/`
8. Manovich, L.: The Language of New Media, pp. 45–46. MIT Press, Cambridge (2001)
9. Lister, M., et al.: New Media: A Critical Introduction, 2nd edn. Taylor & Francis, Abington (2008)
10. Wallace, G.K.: The JPEG still picture compression standard. IEEE Transactions on Consumer Electronics 38(1), 18–34 (1991)
11. JPEG Compression, `http://www.fileformat.info/mirror/egff/ch09_06.htm`
12. JPEG, Wikipedia: The Free Encyclopedia, `http://en.wikipedia.org/wiki/JPEG`
13. TEXT2IMAGE, `http://www.text2image.org`

14. Moere, A.V.: Text2image: Transforming Text into Visual Glitches. Information aesthetics (2009), http://infosthetics.com/archives/2009/02/text2image_transforming_text_into_a_glitch.html
15. Meyer, J.: L'art du texte. Le Matin Dimanche, Lausanne (2009), http://www.lematin.ch/multimedia/coup-coeur/art-texte-87810
16. Hello Bauldoff (2009), http://hello.bauldoff.com/post/182036933/experimental-book-covers-for-a-new-line-of/
17. Cascone, K.: The Aesthetics of Failure: " Post-Digital" Tendencies in Contemporary Computer Music. Computer Music Journal 24(4), 12–18 (2000)
18. Moradi, I., Scott, A., Gilmore, J., Murphy, C.: Glitch: Designing Imperfection. Mark Batty Publischer, New York (2009)
19. HEADer_REMIX, http://www.ffd8.org/header_remix/