Drawing the Glitch
Matthew Austin
Gavin Perin

The introduction of glitches into the production of architectural drawing has the capacity to open up and transform what is understood to constitute digital-architectural production. The architectural drawing uses lines as codified indexical representations of existing or proposed real-world objects. The representation of an edge between a floor and a wall, for instance, requires the line to function through the digitisation of drawings means that they can be easily manipulated drawings according to channels means that lines are no longer the fundamental element of architectural drawing similar to that of algorithmic design but, unlike algorithmic processes, the glitch offers resistance to the representational capacity of a drawing instead of concerning itself with the production of complex forms.

ON THE NATURE OF DIGITAL DRAWING

With the introduction of computer technology into architecture, the hand gestures of drawing a line have been replaced by the pressing of ‘keys’, the clicking of ‘buttons’ and the moving of ‘mice’. The act of drawing is no longer associated with the bodily movements of its traditional production, but is now the job of the algorithm. These algorithms look after the translation from user input to its visual representation in the design process. However, this opens up two important consequences for the drawer, first, there is both temporality and mechanically a fundamental gap between the designer and the visual representation of the drawing on the pixel array. Second, the author has very little control over how the line physically appears on the monitor; a pixel of a monitor or printer change colours as the device gives a digital approximation of the line.

The visual digitisation of the line has translated it from an analogy of a real-world – or at the very least a proposed real-world – object to an analogy in its own right. In this sense, the visual representation of the digital line, and by extension the digital drawing, is constructed from a finite set of numerical values mapping onto an orthogonal pixel array.6 For Matthews,7 this represents an important shift in the nature of drawing as “the discrete, individual nature of each pixel means that the line is no longer the dominant organising principle of image-making”. However, the introduction of the pixel, which is the focus of much curiosity within the study of digital images, highlights an important fissure between digital drawings and pixel arrays; a pixel array can be understood both as a 
\[
m \times n
\]
grid of numerical values mapping onto an orthogonal pixel array or through its linear-sequence representations. In this sense, drawings may not necessarily always be stored on the hard drive as a linear sequence of pixels, but as a series of Cartesian points and geometric constructions around those points. This information is mathematically distorted into ‘view space’ (shown from the perspective of some ‘camera’ which may or may not be orthogonal), then clipped to the viewport (the size of the image the ‘camera’ sees). This abstract mathematical representation of objects is then discretised into two separate pixel arrays (the depth buffer, which in turn helps calculate the final pixel-colouring information) and finally rendered directly onto the pixel array of the monitor. This highlights two crucial points. The first is that a wide variety of algorithms are fundamental to the translation of a drawing moving between the hard or solid-state drive and the pixel array. There is a difference in the way the computer ‘operates’ a vector file compared to a raster file, and there is a further difference in the way that the computer ‘operates’ different types of these files. Different algorithms are used to interpret a drawing for every individual file format; there are algorithms that open .DWS files, algorithms that open .PNG files, algorithms that open .DWG files, algorithms that open .DOC files, etc. These algorithms may transmute the drawing in different ways and thus subtly or significantly create different results upon the pixel array.8 Further, once a digital drawing has been released to its respective audience, it “forestalls the capacity of the author to maintain control over the imaging process”.9 This in turn gives the original author control over what is done with their drawings, but also the software with which they are viewed (i.e. what algorithms are used to translate them from their binary-numeric representation to the pixel representation). The second is that two identical pixel array arrangements may have two drastically different structural representations, as revealed by Fig. 2.

ENTER THE GLITCH

In the early part of this decade, an artist-photographer named Melanie Willhide had her computer, backup drive and external hard drives stolen by Adrian Rodriguez. Rodriguez had wiped the machine and was using it as his own until caught by the local authorities. After the machine was returned to Willhide, she ran recovery software in an attempt to restore her lost work.10 The result was a series of fragmented and distorted copies of her original digital images. In 2012, Willhide exhibited the work in a show in New York titled ‘To Adrian Rodriguez with love’.11 This is a story which offers two important insights for the discussion around digital drawing.

The first is that Mitchell’s assertion that “a digital copy is not a debased descendant but is absolutely indistinguishable from the original”12 is thrown into question. If errors can create the visual surface of the digital image via the very nature of the image being stored on a hard or solid-state drive, then quite equally other modes of storage and transmission can result in debased copies. This should come as no surprise – Shannon highlighted that “since, ordinarily, channels have a certain amount of noise, and therefore a finite capacity, exact transmission is impossible”.13 Here, a channel is considered any medium that has the capacity to transfer information.14 While there are modes of digital transfer between computers (such as email, Dropbox.com and external hard drives), the internal means that the computer transfers the information of a digital drawing from its hard or solid-state drive to RAM, GPUs(s) and CPU(s), as well as transferring
The most prolific and understood form of glitching is the process identified by Davis as ‘data bending’. Data bending is the act of transforming a file’s linear sequence representations, which in turn causes a visual effect. This is frequently done through binary-numeric code, hexadecimal, or ASCII structural representations. An attribute that Broeckmann highlights is that ‘malfunction and failure are not signs of improper production. On the contrary, they indicate the active production of the “accidental potential” in any product’. Virilio says that “the innovation of the ship already entailed the innovation of the shipwreck. The invention of the steam engine, the locomotive, also entailed the invention of derailment, the rail disaster”. The invention of new technology also implies its modes of failure. In the same vein, the file format implies how it renders its failures. It is impossible to give an exhaustive list of data bending as technologies and algorithms shift and change, and file formats are invented, popularised and fall out of use. The way technologies glitch is unique to each medium. Nevertheless, there has already been a study done on how differing image formats glitch. What is of interest here is how digital-architectural production can reconcile such transformations and interpret them spatially.

The second, and more important, point is that this suggests a new method of working with digital drawings, through non-visual and visualised manipulations of a digital drawing’s structural representations. The fetishised application of these techniques is colloquially referred to as ‘glitching’, with the distorted outcomes referred to as ‘glitches’. Gaulon formalises this colloquial definition as follows: “The digital glitch [...] is a way of seeing the code behind a document.” And: “When a digital glitch occurs, it is not the image, the sound or the video that is changed, but their binary code.”

It is worth noting that this definition of what constitutes a glitch is still problematic, as it refuses to engage with important phenomenological and technical issues of definition, highlighted by Moradi and Menkman. However, for the purpose of understanding what the glitch within the nature of architectural drawing constitutes, Gaulon’s more colloquial definition suffices as a mechanism to explore these potentials.

GLITCHING ARCHITECTURE

For the purposes of this paper, a two-dimensional plan of the Barcelona Pavilion is used to visualise the results of a glitch being applied to a digital drawing. The preference for a plan drawing is based on the fact that three-dimensional drawing files are generally quite resistant to transformations because the glitch will likely result in invalid geometry. This is not to say that it is impossible – Mark Klink highlights that the .OBJ file type has this capacity. However, the .OBJ is an ASCII format and as such the information is read by the algorithm as its literal textual interpretation; in other words, a point’s Cartesian coordinates are exactly written in the file as their ‘x’, ‘y’ and ‘z’ values. A further issue is that the operations of manipulating a .OBJ file cannot distort the topology of the geometry, thus making it equivalent to algorithmic distortions available within modelling software. Linear perspective carries with it the issue of literal interpolation. As a mechanism that deals with the ‘void (of meaning)’ created by such a drawing, it is likely to confuse architecture with its image. This is strongly highlighted by Medeguerre Bitnik’s H3333333k, in which the façade of a building is literally transformed to resemble the glitched image. Instead, the sake of clarity, an exploration of the orthographic offers more jarring and difficult questions for architectural drawing in the digital age.

The lack of a clear and singular interpretation of the glitched drawing forces the architect to reconfigure and re-evaluate what these drawings mean spatially. These re-evaluations are not spatially unique. For example, the top-left corner of Fig. 4 acts as an illusion, allowing it to be viewed as a plan with portions skewed or an axonometric (Fig. 5), where the skewed moments in the drawing are vertical projections – however, what the marks on the now-folded surface imply is still unclear. Just as the traditional drawing attempts to narrow the number of valid spatial interpretations through the application of known disciplinary conventions – a property maintained by the surface of traditional digital drawing – glitch drawing disrupts the viewers’ assumed allegory of the images, forcing them to either reject the validity of the image or, more interestingly,
information storage. What does, for example, an interweaving of the Barcelona Pavilion mean spatially? The rotation, translation and scaling of a line or a drawing represents a clear architectural act, as these elements are analogous to an architectural proposition. However, the pixel array represents a line, in as much as its pixels’ RGB values maintain enough contrast with the surrounding pixels and the pixels maintain their position in the array. Because glitch techniques work at the structural level of the image, the extension of the analogy of a line being maintained is not guaranteed. Although these techniques are new in the context of architectural production and an exhaustive investigation would be required to understand the value and nuances specific to each individual one, their value is that they all resist the very thing the drawing purports to represent.

It is evident that the glitching of a plan requires a complete reconsideration of the vertical nature of the result, and in turn the glitching of an elevation requires a readjustment of the plan. In fact, the glitch not only resists architectural convention, but also disrupts the relationship between architecture’s different modes of representation. Further to this, architecture’s other modes of representation (such as video) constitute a difference in technology and thus glitch in a fundamentally different way. The glitch has the potential to disrupt architecture at any point within its production to force a complete reworking of what the architectural drawing intends to represent.

Where traditional modes of digital drawing shift the line as the predominant organising structure of the drawing to the pixel, the glitch shifts it from the pixel array to the unfamiliarity of non-visual representation. The drawing’s hidden binary-numerical nature and polymorphism unite with the nature of digital media to offer architecture the capacity to resist its own disciplinary conventions.

In this sense, the glitch of a drawing demands a reimagining of the grammatical assumptions of our representations. Instead of attempting to close down the interpretation of the drawing into a single unique spatial condition, the glitch denies the viewer this opportunity and is therefore dependent upon the individual’s capacity to interpret and spatially reconcile a reworking of the surface representation of the surface of the drawing originally represented.

REFERENCES


ACKNOWLEDGEMENTS

The authors would like to acknowledge Kieran Patrick for the production of the Barcelona Pavilion plan used for the glitching process.