

Procesos en la inmaterialidad. El caso del *databending*

David Baulina Guzman

Arte e Investigación (N.º 16), e039, noviembre 2019. ISSN 2469-1488

<https://doi.org/10.24215/24691488e039><http://papelcosido.fba.unlp.edu.ar/ojs/index.php/aei>

Facultad de Bellas Artes. Universidad Nacional de La Plata

La Plata. Buenos Aires. Argentina

PROCESOS EN LA INMATERIALIDAD

El caso del *databending*

PROCESSES IN THE IMMATERIALITY

The Case of Databending

DAVID BAULINA GUZMAN

davidbaulina@hotmail.com

Facultad de Bellas Artes. Universidad Nacional de La Plata. Argentina

Recibido 18/5/2019 | Aceptado 14/9/2019

Resumen

Una cuestión clave en los procesos creativos de la cultura material es la adquisición de habilidades a través de la experiencia y del oficio. En el trabajo creativo que se realiza en artes visuales, gran parte de estas habilidades y experiencias nacen del mismo material desde el que se trabaja. Sin embargo, en el caso del trabajo con medios digitales, esta materialidad se presenta como exigua o nula. Las imágenes digitales generan una situación nueva con respecto al trabajo que se ejercía anteriormente con medios analógicos. Como caso de estudio abordaremos el de la elaboración de imágenes digitales formadas por un proceso conocido como *databending* o rotura de archivo de imagen.

Palabras claveInmaterial; forma; *glitch***Abstract**

A key issue in the creative processes of material culture is the acquisition of skills through experience and craft. In the creative work that is done in visual arts, a large part of these skills and experiences is born from the same material from which one works. However, in the case of work with digital media, this materiality has an unsubstantial nature. Digital images generate a new situation with respect to the work previously performed with analogue media. As a case study we will address the development of digital images formed by a process known as *databending*.

KeywordsUnsubstantial; form; *glitch*

Esta obra está bajo una Licencia Creative Commons Atribucion-NoComercial-CompartirIgual 4.0 Internacional

Para poder hablar de qué es *databending*, primero es necesario remitirse al *glitch*, del cual este es solo un proceso particular. El *glitch* es la práctica por la cual errores en la formación de imágenes digitales son buscados ex profeso, como un resultado en sí mismo y no como alteraciones que deban ser descartadas. El concepto viene originalmente de la electrónica, campo en el que era usado para referirse a un pico o cambio de voltaje en una corriente eléctrica (Kozak, 2012). Ya en el ambiente de la programación o de los medios digitales comenzó a usarse con el sentido con el que lo conocemos actualmente, como un error que provoca algún tipo de alteración —en el funcionamiento del programa o en la imagen o video digital— sin llegar a provocar la caída del programa o la ilegibilidad de la imagen, caso conocido normalmente como *bug*. Las diferencias entre uno y otro concepto suelen variar. En el mundo del desarrollo de videojuegos puede encontrarse al *bug* como un error en la programación y al *glitch* como la expresión de ese error, pero la mayoría coincide en poner a ambos del lado de los efectos de esta mala sintaxis. Mientras el *bug* suele producir un error masivo y una caída del programa, el *glitch* suele tener efectos menos negativos, limitándose a efectos de sonido o audio.

Llegados a este punto, nos encontramos con dos casos muy diferentes de *glitch*. Por un lado, el puro, es decir, el error imprevisto, y por el otro, el producido por una decisión deliberada, es decir, por un sujeto que manipula determinados artefactos con el fin de que estos devuelvan un error. El primer caso es accidental, un hecho fortuito o un hallazgo, y es producto de un error genuino. El segundo es deliberado, diseñado para que devuelva este error, y es producto de una artimaña, del suficiente conocimiento de los resortes de una máquina como para poder torcerlos. Nosotros nos ocuparemos de este segundo caso. Las situaciones que pueden dar lugar a estos errores son variadas, entre las que se cuentan el daño físico al soporte de un archivo de imagen (rayadura de CD, DVD o disquete) o los daños en el archivo producidos por errores en la conversión de formato o en la compresión (errores en videos subidos a internet, mal manejo de la conversión de un formato de video a otro). El error puede producirse, entonces, a la altura del dispositivo que lo reproduce o a la altura del código mismo que soporta al medio. El *databending* pertenece a este último grupo.

De este modo, el *databending* se produce como un error en el código del archivo, fenómeno que puede darse de tres formas distintas: por editar incorrectamente (es cuando se edita un archivo usando programas destinados a otro tipo de archivos); por reinterpretar (cuando un archivo es convertido de un formato a otro); por forzar errores (cuando se usan *bugs* de los programas para forzar la rotura del archivo de imagen).

De los tres modos de producir esta rotura, el que da una mayor ratio es el de la reinterpretación, ya que recibe modificaciones menos profundas. Este será el caso que utilizaremos. Consiste en hacer interpretar como datos crudos (RAW) un archivo por algún programa que soporte otros formatos, es decir, no diseñado para ese tipo en particular, y luego de esto (pudiéndose modificar el archivo o simplemente pasándolo de un formato a otro), volver a armarlo con su formato original o conservando este nuevo formato.

También puede realizarse al fabricar un programa, algún *script* sencillo que modifique los datos en el archivo original. Como ejemplo, se puede hacer leer un archivo de imagen comprimido en formato JPG por un programa de edición de sonido (Audacity o Ableton), que lo interprete como datos de audio crudo, guardarlo en ese programa también con formato de datos crudos (RAW) y posteriormente hacerlo interpretar por un programa de edición de imágenes (como Photoshop o Gimp). Ya en este simple proceso el archivo original se modifica y devuelve, en el mejor de los casos, una imagen diferente a la que se introdujo o, en el peor de los casos, un archivo imposible de ser leído por ningún programa, perdiéndose la imagen.

A continuación, se ha dividido la descripción del proceso de rotura en dos momentos (I y II) que coinciden, parcialmente, con un momento más definido por el material (I) y uno más definido por la técnica (II).

I

Suele darse una menor tasa de pérdida de las imágenes en formatos con poca o ninguna compresión, esto es, archivos a los que no se les han borrado píxeles que se consideraban superfluos. Por ejemplo, el formato RAW (crudo) o alguno con compresión sin reducción de la calidad de imagen (*LossLess*) como el TIFF. La razón de esto se debe a una modificación en el espacio de color de la imagen. La compresión significa un cambio sustancial en el espacio de color que utiliza la imagen. El espacio de color es un sistema que ordena los colores de algún modelo de color (por ejemplo, RGB o CMYK). Este espacio de color puede ser más o menos completo. Estos espacios asignan a un mismo píxel tres valores: uno para el rojo, otro para el azul y otro para el verde. En el caso del JPG el espacio de color se organiza de otro modo. En vez de poseer tres canales de color, asigna a cada píxel un canal de brillo y dos de color (el canal azul menos el de brillo y el canal rojo menos el de brillo). Aquí empieza a intervenir otro proceso que es el submuestreo de croma, determinante en la economía de la información en la imagen. No ahondaremos en este problema, pero incluso si supusiéramos que se le asigna un submuestreo que en la práctica no significara ninguna compresión, la información de los píxeles ya de por sí se habría visto afectada, no en su cantidad sino en su estructura. Por supuesto, el formato JPG comprime de otros modos la información de la imagen (suaviza bordes y los desenfoca), pero esta pérdida de información no parece tan problemática como el cambio de estructura que se genera. Esto nos lleva al siguiente problema, el formato de los datos de imagen. El mismo espacio de color RGB puede tener distribuidos sus canales en básicamente dos formas: empaquetados o de forma planar. Empaquetado significa que los tres valores (rojo, azul y verde) están guardados juntos, mientras que en el formato planar cada uno de los canales está guardado por separado. En general, las imágenes empaquetadas producen solamente ruido blanco como resultado, siendo más probable de recuperar aquellas que guardaron cada canal por separado.

Finalmente, en el caso de querer modificar el archivo, ya sea con un editor de texto o de sonido, usualmente es necesario no realizar cambios en el encabezado o en la parte final ya que poseen información sobre cómo debe leerse. La modificación de estas áreas del código suele traer aparejada la destrucción de la imagen. Al mismo tiempo, estos trozos de código inalterado suelen verse reflejados como bandas horizontales (una superior y una inferior) discontinuas con respecto al resto. Esto puede obviarse en el caso de trabajar con formatos RAW (crudo) ya que no poseen un encabezado.

Estos pocos ejemplos grafican los problemas con los que puede enfrentarse uno al trabajar con este tipo de materialidades. Siempre que se trabaje a través del espacio de las interfaces, los archivos están permanentemente ocultando el código que los forma y los estructura. Sin embargo, este tipo de prácticas revela la preponderancia del código por sobre la imagen.

Lo que observamos en las imágenes digitales no es más que la sombra del código que las motiva. Fenómenos como el *glitch* hacen visible esta materialidad. Al manipular una imagen digital lo que está modificándose realmente es su código y este no es algo realmente visible: «La imagen digital es un efecto de la visualización del archivo de imagen, que es invisible [...]» (Groys, 2016, p. 161). Hablar de materialidad en contextos digitales resulta complejo por las características del medio. Suele usarse, por el contrario, su contraparte, esto es, la inmaterialidad. Al proceso de digitalización en el que estamos inmersos se lo asocia con uno de desmaterialización, esto es, un proceso de «[...] contracción del mundo de los objetos materiales, objetos que serían sustituidos por procesos y servicios cada vez más inmateriales» (Maldonado, 1999, p. 13). Sin embargo, no parece que en el mediano plazo el contacto con un soporte material, aunque exiguo, vaya a desaparecer. A esta materialidad rala se la carga progresivamente de mayores volúmenes de información. En el caso de las imágenes digitales este aumento de la información opera bajo la forma más evidente de una mayor complejidad (mayor grado de iconicidad, por ejemplo) de la imagen. Pero este aumento de información también vale en el sentido de una aparición de lo menos probable, es decir, en un sentido de la teoría de la información (Flusser, 2015, p. 12).

En el contexto de la práctica artística puede sernos de utilidad acudir al concepto de *material artístico* que desarrolla Theodor W. Adorno (2013) a lo largo de su pensamiento estético. El material artístico para Adorno es un material que se ha extraído de la naturaleza y del que se ha extraído su forma original mediante el trabajo de la técnica. Este material no debe confundirse con algo matérico: el concepto de material de Adorno no es sustancialista. En este sentido, puede ser esclarecedor un pasaje de su monografía sobre Alban Berg: «el material es aquello con lo que los artistas juegan: las palabras, los colores y los sonidos... también las formas pueden ser material» (Adorno, 1990, p. 199). Por supuesto que Adorno está pensando en experiencias como el dadaísmo o el constructivismo, experiencias que habían corrido la frontera del dominio estético. Planteos como «Del concepto de material parece haberse tomado consciencia en los años 20» (Adorno, 1990, p. 200) apuntan hacia

esta ampliación, durante las primeras vanguardias, del conjunto susceptible de ser tratado como material artístico. Más allá de las consecuencias más evidentes de este concepto de materialidad (expandido hasta poder abarcar fenómenos de una nula sustancialidad), el corolario del concepto de material artístico de Adorno se verá también en la relación de este material con la técnica.

II

En el punto en el que habíamos dejado la práctica del *databending*, el proceso consistía en no perder los archivos de imágenes. Una vez logrado un flujo de trabajo continuo, este puede volverse monótono y llegar a resultados cada vez más previsibles. Todos los métodos relacionados con el *glitch* llevan a roturas o a ruido en la imagen, que entran en un número reducido de categorías:

- Fragmentación: hace referencia al proceso en el que la imagen se recorta en porciones ortogonales y estas son desplazadas en el cuadro de la imagen. Suele ser en trayectoria horizontal debido al sentido de lectura de la imagen por parte de las computadoras.
- Repetición: es cuando la imagen es clonada y se itera, generalmente, en formato más pequeño, dentro del marco original.
- Alineado: algunos o todos los elementos de la imagen se descomponen en fragmentos siguiendo líneas (de nuevo, en dirección horizontal). Esto puede darse en los píxeles o en los planos de color.
- Complejidad: supone la destrucción completa de la imagen, formando patrones, sin llegar al nivel de convertirse en un puro graneado o en un ruido blanco.

A estas categorías, ya propuestas por Iman Moradi (2004) en su artículo sobre el *glitch*, podría incluirseles:

- Cambio de tonalidad: cuando una parte o la totalidad de la imagen cambia de gama de colores o adquiere alguna tonalidad plana.
- Manifestación de la compresión: ocurre con espacios de color que no siguen un esquema RGB (asociado al submuestreo de croma de la imagen).
- Manifestación del código: más infrecuente en el *databending* que en otros tipos de *glitch*. Es justamente la manifestación del código de la imagen, en forma de palabras, letras, números o trozos de código en algún lenguaje de programación.

En general, el resultado del proceso es una mezcla de estas características. A medida que las modificaciones aumentan (en cantidad, en grado), la imagen va destruyéndose y se hace cada vez más difícil de leer por algún programa. Eventualmente, y debido al grado de distorsión, terminará dando un resultado similar al ruido blanco. En los casos que no se desee la total destrucción de la imagen, el artista debe intentar una modificación del código a menor escala. Entre los recursos que pueden utilizarse están los siguientes:

- Formatos sin compresión: más allá de lo dicho anteriormente sobre el éxito en la empresa de generar el *databendig*, los formatos sin compresión producen un ruido en la imagen menos invasivo que los formatos comprimidos, en los que suelen aparecer motivos similares a grillas.
- Formato de archivo: al momento de hacer leer un archivo por un editor (destinado para otro tipo de datos) puede modificarse el mismo con las herramientas de este programa. Estas alteraciones suelen dar los resultados más espectaculares, pero también agregan ruido a la imagen. Minimizar el uso de estos recursos previene a la imagen de retoques excesivos.
- Espacio de color planar: el espacio de color planar se relaciona con la división en planos de los canales de luz RGB. En vez de mezclarse los píxeles de los tres canales, estos permanecen separados en capas distintas. Esto produce una modificación mucho menos caótica de la imagen.
- Tamaño de imagen: el proceso suele producir menos ruido en imágenes de gran tamaño (gran cantidad de píxeles) y gran resolución.
- Render: imágenes creadas directamente en un entorno virtual (un programa de modelado en 3D, por ejemplo). Suelen dar un mejor resultado que las fotografías. Esto se debe a dos razones distintas. Por un lado, las imágenes creadas virtualmente poseen, incluidas aquellas con un grado de iconismo muy realista (rénderes para arquitectura, por ejemplo) una información menos caótica que las fotográficas. Por el otro, el trabajo con formatos digitales permite un grado de elección sobre el formato del archivo que difícilmente posean las fotografías.

Como vemos, la técnica está destinada a producir en el archivo modificaciones que se muevan entre el punto de la no lectura de la imagen y el de la disolución total de la misma en algún tipo de ruido. Volviendo a la estética adorniana, la técnica es el nombre estético que se da al dominio del material (Adorno, 1990, p. 280). Adorno ve a los materiales y a la técnica con la que se trabaja como históricamente determinados, en el sentido del dominio progresivo del material y del refinamiento de la técnica. Sin embargo, el concepto de técnica adorniano no refiere a un sentido clásico de refinamiento artesanal (sentido que Adorno encontraba en el arte premoderno), sino que está determinado por su propio material. La técnica en el arte no debería ser algo que se le impusiera al material por fuera de sí, como algo que le fuera ajeno, sino que es el material mismo el que exige su técnica específica. Esto pareciera cobrar sentido el trabajar en archivos digitales al nivel de su código.

Conclusiones

El proceso del *databendig* consiste en una modificación sobre imágenes digitales realizada al nivel de su código. Este proceso puede volver evidente la especificidad de las imágenes digitales: que no son tanto imágenes con un código, sino un código que es susceptible de ser leído como una imagen.

La forma que adquieren estos archivos cambia continuamente y esto no depende de casos tan espectaculares como el del *databending*. Un mismo archivo proyecta realmente distintas imágenes dependiendo del programa utilizado para interpretarlas, sin la necesidad de la rotura de su código para que esto ocurra. En el caso del *databending* esto supone, para un mismo archivo, la no lectura por parte de un programa, la lectura solo como ruido o la lectura de formas completamente disímiles por parte de uno u otro programa. Esta disparidad le da más peso a la idea de que sobre lo que se trabaja es, en última instancia, un código y no una imagen.

Si la forma de este código (o la falta de aquella) depende de un programa, significa que depende de un trabajo de interpretación (Groys, 2016). Parece tejerse un paralelismo con la composición de partituras en la música y su ejecución. La interpretación está, en gran medida, librada al ejecutante. En este caso, el trabajo del artista se sitúa al nivel de la composición del código y la interpretación de su forma queda a cargo de una máquina.

Por supuesto, una vez logrado el efecto o la imagen deseada puede pasarse a un formato más estable. Quizás la forma más exitosa de lograr esto sea simplemente realizar una captura de pantalla del resultado obtenido. Pero de esta manera se desactiva el artefacto, generándose una especie de instantánea, de forma fosilizada o estática del archivo. La potencialidad del *databending*, parece ser la de generar este artefacto capaz de actualizarse permanentemente.

Si retomamos la teoría de Adorno, la forma no es un principio atemporal, sino que es contenido sedimentado, refleja las condiciones históricas en las que fue producida (Buck-Morss, 1981). Adorno veía en la atonalidad (y en las obras de arte inorgánicas) un reflejo del período histórico presente, un período que percibía como de decadencia burguesa. El progreso sobre el dominio del material, el progreso de la técnica, solo lo sería en el sentido de una progresiva desintegración.

El trabajo sobre el código de archivos de imagen parece ir, como técnica, en esta dirección. Por supuesto, este recurso por sí solo no podría definir el estado de actualidad o de avance de una obra, estado que Adorno definía como algo que debería producirse en todas las dimensiones de la obra al mismo tiempo (Adorno, 2003), pero podría apuntar a algún tipo de actualidad en, al menos, una dimensión del material, esto es, de la práctica sobre las imágenes digitales.

Referencias

Adorno, T. W. (1990). *Alban Berg: El maestro de la transición ínfima*. Madrid, España: Alianza.

Adorno, T. W. (2003). *Filosofía de la nueva música*. Madrid, España: Akal.

Adorno, T. W. (2013). *Estética (1958/59)*. Ciudad Autónoma de Buenos Aires, Argentina: Las Cuarenta.

Buck-Morss, S. (1981). *Origen de la dialéctica negativa*. Ciudad de México, México: Siglo Veintiuno.

Groys, B. (2016). *Arte en flujo*. Ciudad Autónoma de Buenos Aires, Argentina: Caja Negra.

Flusser, V. (2015). *El universo de las imágenes técnicas*. Ciudad Autónoma de Buenos Aires, Argentina: Caja Negra.

Kozak, C. (2012). *Tecnopoéticas argentinas*. Ciudad Autónoma de Buenos Aires, Argentina: Caja Negra.

Maldonado, T. (1999). *Lo real y lo virtual*. Barcelona, España: Gedisa.

Moradi, I. (2004). *Glitch Aesthetics*. Recuperado de <http://www.organised.info/wp-content/uploads/2016/08/Moradi-Iman-2004-Glitch-Aesthetics.pdf>