

Compresión: entre árboles y símbolos

Sigifredo Escobar Gomez
Yuli Paola Vargas Rodríguez
Maria Camila Garcia Parada
Ricardo Cedeño Montaña

sigifredo.escobar@udea.edu.co
Grupo de Investigación-Creación Contracampo,
Semillero Imagen Algoritmo
Universidad de Antioquia
Medellín, Chile

Resumen

Los algoritmos de compresión son una de las piezas esenciales de la cultura digital; su uso es notorio en formatos digitales de imagen y audio como JPEG, GIF y MP3. Aunque es cotidiana la circulación de estos archivos, los principios técnicos que hacen posible la compresión digital rara vez se exploran fuera de las ciencias de la computación. Esta ponencia presenta un proceso de investigación-creación sobre la implementación de estos principios como parámetros compositivos para la creación de imágenes digitales. Dos aproximaciones a la compresión son consideradas aquí: (a) como mecanismo para la reducción de redundancia y (b) como la combinación de la cantidad mínima de elementos para producir objetos gráficos complejos. Así, el proceso de creación comprende por un lado la implementación de dos algoritmos históricos: los árboles de Huffman y las tablas de Level-Ziv-Welch (LZW) y, por otro lado, el lenguaje gráfico usado por las etnias Wichi (Chile) y Arhuaco (Colombia) en tejidos-pictogramas. El resultado de combinar procesos computacionales con un sistema de símbolos gráficos étnicos expande las posibilidades visuales usando la compresión como principio general para la creación. Las formas resultantes permiten reflexionar sobre la compresión como uno de los principios modeladores de la cultura digital contemporánea.

Palabras Clave

Algoritmo, compresión, creative coding, símbolo Arhuaco, símbolo Wichi, GIF, Árbol de Huffman

Introducción

Los algoritmos de compresión tienen origen en la necesidad de disminuir la cantidad de espacio usado por los datos almacenados de archivos con diferentes formatos. Para lograr una transmisión más veloz, la codificación de estos puede darse en dos vías (a) con pérdida (Lossy) reduciendo los bits al eliminar información menos importante o innecesaria y (b) sin pérdida de información (Lossless), disminuyendo los bits al identificar la redundancia estadística. En el campo de las imágenes digitales la compresión retoma el problema sobre la reducción de datos necesarios para representarlas y se enfoca en la disminución de aquellos que son redundantes [1].

El uso cotidiano de la compresión se hace notorio cuando por ejemplo los servicios de correo electrónico alertan sobre la superación del peso máximo de archivos a enviar, en este caso acudimos a programas del sistema que disminuyen su tamaño mediante el formato ZIP. Para los archivos de audio es posible reducir el espacio de almacenamiento con su conversión a formato MP3 ya que son recortadas las señales fuera del rango de audición humana. Adicionalmente, encontramos el uso de los algoritmos de compresión en prácticas creativas como el Glitch art, género artístico que explora las posibilidades estéticas de los errores en las piezas visuales, es decir, las fallas digitales se toman como una característica intrínseca de las obras.

Esta ponencia está dividida en cinco partes. La primera parte, presenta el problema central de la compresión y nuestra aproximación al mismo. En la segunda parte presentaremos un marco teórico donde discutiremos conceptos básicos de compresión como los árboles de Huffman y las tablas de color en el formato GIF. A modo de metodología de investigación-creación, la tercera parte presenta: el flujo de trabajo, el análisis para el diseño de una colección de 16 símbolos y el proceso de combinación de estos símbolos a partir del principio de compresión de información ideado por David Huffman en 1952 [2]. En la cuarta parte, presentamos los resultados de creación en forma de un archivo de imagen GIF animado generado a partir de textos con el himno nacional de Colombia. Finalmente, discutiremos algunas conclusiones a partir de los códigos y las imágenes generadas.

Planteamiento del problema y justificación

El acercamiento conceptual a los fundamentos algorítmicos de la compresión y sus variaciones nos permite advertir nuevas formas compositivas resultantes de la alteración de atributos técnicos durante la producción de imágenes. Las piezas resultantes de este proceso de investigación-creación no responden

necesariamente a los propósitos prácticos de la compresión digital: poco almacenamiento y ágil transmisión, sin embargo, son consecuencia de sus principios técnicos. Tras la implementación de procesos computacionales y la alteración de un atributo como la tabla de colores, se amplía el universo de imágenes digitales. Las formas resultantes permiten reflexionar sobre la compresión como uno de los principios modeladores de la cultura digital contemporánea.

Otro aspecto relevante abordado por este proyecto es que la compresión no se limita al mundo de la computación. Las imágenes y lenguaje gráfico de la mayoría de las comunidades indígenas tiene entre sus principios el menor uso de elementos gráficos que posteriormente componen complejas figuras usadas en tejidos y pictogramas, esta es otra forma de compresión que este proyecto considera.

El proceso de trabajo inicia estudiando los árboles de frecuencias de Huffman y las tablas de color en los algoritmos Level-Ziv-Welch (LZW)[3]. Así, iniciaremos primero describiendo los objetivos que nos planteamos en este proyecto de investigación-creación.

Objetivos

Identificar parámetros visuales para el desarrollo de un código de programación para la creación de imágenes digitales, cuyo principio se fundamente en dos algoritmos de compresión históricos: los árboles de Huffman y las tablas LZW.

Desarrollar una implementación propia del algoritmo de Huffman para la compresión de cadenas de caracteres a partir de sus frecuencias.

Crear una colección de símbolos gráficos inspirados en las representaciones visuales usadas por las etnias Wichi (Chile) y Arhuaco (Colombia).

Escribir un código de programación que implemente las modificaciones realizadas al principio algorítmico de compresión de las tablas LZW, tomando como insumos de entrada la colección de símbolos gráficos.

Realizar una instalación que proyecte las animaciones tipo GIF producto del código desarrollado.

Marco teórico

Huffman: los árboles

Para enfrentarnos a este problema de creación, tomamos como punto de partida el algoritmo propuesto por David Huffman en 1952 para reducir el tiempo de transmisión de secuencias de

símbolos. Esta decisión se fundamenta en que este algoritmo histórico es uno de los que tiene más derivaciones en métodos de compresión. Su principio se basa en la recodificación de cada bit en función de la cantidad de veces que un carácter aparece en una secuencia de caracteres, es decir, a partir de su frecuencia [2]. Huffman lo hace mediante la construcción de un árbol binario que a su vez contiene árboles o nodos intermedios, contruidos tras la suma de frecuencias de los nodos hoja. Revisemos el ejemplo que Thomas Cormen presenta con una cadena de ADN en la que cada aminoácido es considerado como un carácter (A, C, G y T) al que se le asocia un porcentaje según su frecuencia en la cadena [4], ver figura 1:

El algoritmo de Huffman toma entonces los nodos hoja con menor frecuencia (C y G) y los suma para formar el nuevo nodo intermedio cuya raíz será el resultado de la operación. De esta manera se sigue hasta que no haya nodos hoja por unir.

Como se puede observar en la figura 2 a los vértices del lado izquierdo se les asocia el 0 y a los del lado derecho 1, el código Huffman se construye entonces tomando los dígitos binarios del recorrido iniciando por la raíz del árbol hasta cada una de las hojas, para el ejemplo A: 0, C: 100, G: 101, T: 11.

El GIF: las tablas de color

El segundo componente estudiado es el algoritmo de compresión usado en la codificación Lempel-Ziv-Welch (LZW) que data de 1977 y cuyo uso se extiende a formatos de archivos de imagen como GIF, PDF y TIFF [1, p. 564]. Este algoritmo tiene concordancia con el principio desarrollado por Huffman pero ahora aplicado a mapas de bits. Sin embargo, se diferencia en que al ser un algoritmo adaptativo, LZW lee el flujo de datos y de manera inmediata genera una tabla de códigos -o diccionario-, para seleccionar secuencias de datos desde la primera lectura del archivo comprimido [3].

El problema de compresión en las imágenes digitales es lograr de forma eficiente que “una computadora muestre una imagen y al mismo tiempo ahorre memoria” [5]. El formato GIF logra esto a través de una variación del algoritmo LZW que guarda en cada pixel una tabla de colores que contiene sus índices y valida si hubo un cambio en esta o no [6]. A continuación, quisiéramos entrar en dos detalles del funcionamiento del formato de imagen GIF: los diccionarios y los bloques. Esto tiene como propósito ilustrar la manera en que el formato genera la estructura de datos que darán forma a la imagen, permitirá validar si hay o no repetición de información y posibilitará la posterior decodificación del archivo comprimido.

Primero observemos el uso de diccionarios en el formato GIF. Esta es una estructura que contiene una tabla básica de colores junto a un código numérico asignado a cada uno de ellos. Lo que hace

el algoritmo es ir agregando, en la tabla inicial, las combinaciones que van apareciendo a medida que se lee la imagen, esto, para tener identificados y nombrados colores que pueden repetirse en el flujo de datos y así disminuir la codificación al reutilizar el código numérico que se ha ido generando en la tabla de colores. Para el caso de los bloques, Kuratomi explica que la estructura general de este formato está compuesta por un encabezado que representa el formato de archivo GIF con códigos en caracteres ASCII -hexadecimal- y la versión que se utiliza para codificar la imagen (87a, 89a) [6]. El siguiente bloque es el descriptor de pantalla lógica, que le indica al decodificador el espacio que ocupará la imagen: ancho y alto del lienzo, el byte 91 es un descriptor que se expande a binario como 10010001, cada dígito da información sobre si la tabla de colores es global, indica la resolución del color y la relación del aspecto de los píxeles. El bloque que sigue representa la tabla de colores global que "consta de una lista de intensidades de componentes de color RGB (rojo-verde-azul), tres bytes para cada color, con intensidades que van de 000000 (mínimo) a FFFFFFFF (máximo) en hexadecimal. El color (00, 00, 00) es el negro más profundo, el color (FF, FF, FF) el blanco más brillante. Los otros colores extremos son rojo en (FF, 00, 00), verde en (00, FF, 00) y azul en (00, 00, FF)" [6], ver figura 3.

La tabla continua con los siguientes bloques: el de extensión de control de gráficos para las animaciones y la configuración de la transparencia en el color; enseguida el descriptor de imagen donde se tiene información sobre la cantidad de imágenes, posición, dimensiones, tabla de colores en caso de ser diferente a la global y una bandera que entrelaza las imágenes para reducir el efecto visual de superposición. Finalmente, se ubica el bloque de datos de imagen real cuyos códigos de salida le indican al decodificador qué color ubicar en el lienzo de acuerdo a la distribución por píxeles; el último dato de la tabla es siempre 3B ó un EOL (el final del archivo).

Metodología

La dinámica inicial tiene un carácter iterativo no lineal con retroalimentación y procesos paralelos desarrollados por los integrantes del equipo de creación. Partimos de una implementación propia del algoritmo desarrollado por Huffman para la compresión de cadenas de texto en el lenguaje de programación Python. De este se obtienen códigos binarios Huffman que serán agrupados en cadenas de cuatro dígitos binarios. Las posibles combinaciones entre 0 y 1 para cadenas de cuatro dígitos son 16 y este es el número de símbolos que se seleccionan a partir del principio modular mencionado por David Consuegra en su estudio acerca del cuadrado [7]. Además del principio modular, evaluamos el grado de variabilidad, es decir el alcance compositivo de cada una de las piezas al combinarlas con las otras.

Los símbolos se vinculan con un relato tomado de las representaciones visuales de la etnia indígena Wichí, asentada

en América del Sur, cuyas referencias primarias tienden hacia las plantas, los animales y el ecosistema que habitan en general. Estas imágenes son los datos que conformarán la estructura tomada de las tablas de color (referenciadas del formato GIF) y que ocuparán el lugar de los colores: Generamos entonces una colección de 16 imágenes que presentamos en la tabla 1 a partir de una retícula base de 17 filas x 17 columnas.

Al tener seleccionados los 16 símbolos derivados del estudio de las mochilas arhuacas presentado por Hernan Rafael Mariotte Peña [8], asignamos una nomenclatura, de acuerdo a la cantidad de píxeles negros que éstas contienen, de manera ascendente iniciando con 0000 y finalizando con 1111 para construir una colección que contendrá los bloques de información y los símbolos.

El programa que escribimos en Python crea un archivo con formato GIF a partir de una cadena de caracteres e identifica la frecuencia de estos a lo largo del archivo, usando nuestra implementación de los árboles de Huffman. El resultado de este paso es entonces un árbol binario que contiene las frecuencias de todos los caracteres. Usando la colección de símbolos y sus códigos binarios, ver tabla 1, la segunda cadena de operaciones del programa se encarga de componer la imagen. El archivo resultante es la superposición de una imagen sobre otra, producto de la sustracción de negros, ver figura 4.

Símbolo	Código binario	Símbolo	Código binario
	0000		1000
	0001		1001
	0010		1010
	0011		1011
	0100		1100
	0101		1101
	0110		1110
	0111		1111

Tabla 1: Símbolos inspirados en la etnia Wichí con su respectivo código binario.

Resultados

A continuación, presentamos una breve descripción del código desarrollado en Python. La figura 5 presenta un diagrama flujo con el funcionamiento del código desarrollado. El algoritmo recibe como entrada una cadena de texto sin un tamaño límite, la función leer texto. Con base en esta, se crea el árbol de Huffman. Con el árbol ya creado, tomamos de nuevo el texto original y encontraremos el código Huffman de cada palabra. Este código lo dividiremos en cadenas de cuatro números, las cuales representan un número binario dentro de nuestra base de datos de símbolos.

Ya teniendo este conjunto de cadenas, el cual representa la palabra dada, las mezclamos usando como base el principio de la compuerta XOR ilustrado en la tabla 2. Es decir, recorremos cada pixel de cada imagen y las vamos mezclando. Dando como resultado una imagen con unos valores en blanco y en negro que representan la palabra.

Entrada		Salida
Blanco	Blanco	Negro
Blanco	Negro	Blanco
Negro	Blanco	Blanco
Negro	Negro	Negro

Tabla 2: Tabla de verdad adaptada de la compuerta XOR

El anterior procedimiento lo hacemos para todas las palabras en el texto. Finalmente, teniendo el conjunto de imágenes que representan el texto, pasamos a darles un tiempo dentro de la animación, esto lo hacemos basados en la frecuencia en la que encontramos cada palabra en el texto. A la palabra con mayor frecuencia, le damos un tiempo de cinco (5) segundos, y a la palabra con la menor frecuencia, le damos un tiempo de 0.1 segundos. Con este tiempo de duración y estas imágenes, creamos un GIF animado a partir del texto completo del himno de la República de Colombia. A cada estrofa corresponde un archivo GIF; los resultados pueden ser vistos en la siguiente dirección electrónica: <https://github.com/Semillero-IA/compresion-resultados>

Conclusiones

La producción de imágenes digitales se encuentra mediada en la mayoría de los casos por un interés publicitario, técnico o

funcional. En el caso de las prácticas relacionadas con el creative coding es posible hacer uso de herramientas ya existentes, reconstrucciones de algoritmos históricos y vincularlas con la cultura visual de los actores que las generan. En nuestro caso, la posibilidad de generar puentes entre algoritmos computacionales implementados de tal forma que son transparentes para los usuarios finales y formas representativas indígenas sin duda es muestra del rango expresivo con el que contamos en la actualidad al vernos atravesados por elementos técnicos a nivel de software y los rasgos interculturales que definen nuestra identidad.

Referencias

- [1] Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing (4.a ed.). Pearson.
 - [2] Huffman, D. A. (1952). A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9), 1098-1101. <https://doi.org/10.1109/JRPROC.1952.273898>
 - [3] Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3), 337-343. <https://doi.org/10.1109/TIT.1977.1055714>
 - [4] Cormen, T. H. (2013). *Algorithms Unlocked*. MIT Press.
 - [5] Boissoneault, L. (2017, junio 2). A Brief History of the GIF, From Early Internet Innovation to Ubiquitous Relic | History | Smithsonian Magazine. A Brief History of the GIF. <https://bit.ly/3BJxp9y>
 - [6] Kuratomi, T. (2019). What's In A GIF [GIT]. The GIFLIB project. <http://giflib.sourceforge.net/whatsinagif/index.html>
 - [7] Consuegra, D. (1992). En busca del cuadrado. Univ. Nacional de Colombia.]
 - [8] Mariotte Peña, H. R. (2020, febrero 6). Significados de las Mochilas Arhuacas | Artesanías Colombianas Online. <https://www.artesaniascolombianas.online/significados-de-las-mochilas-arhuacas/>
- Bibliografía**
- Boissoneault, L. (2017, junio 2). A Brief History of the GIF, From Early Internet Innovation to Ubiquitous Relic | History | Smithsonian Magazine. A Brief History of the GIF. <https://bit.ly/3BJxp9y>
- Consuegra, D. (1992). En busca del cuadrado. Univ. Nacional de Colombia.

Cormen, T. H. (2013). *Algorithms Unlocked*. MIT Press.

Fuller, M. (2008). *Software studies: A lexicon*. MIT Press.

Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4.a ed.). Pearson.

Huffman, D. A. (1952). A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9), 1098-1101. <https://doi.org/10.1109/JRPROC.1952.273898>

Kuratomi, T. (2019). What's In A GIF [GIT]. The GIFLIB project. <http://giflib.sourceforge.net/whatsinagif/index.html>

Mariotte Peña, H. R. (2020, febrero 6). Significados de las Mochilas Arhuacas [Son 16 Diseños] Estos Son: Artesanías Colombianas Online. <https://www.artesantiascolombianas.online/significados-de-las-mochilas-arhuacas/>

Montani, R. (2018). Imágenes indígenas del bosque chaqueño: Animales y plantas en el universo visual wichí. *Caravelle. Cahiers du monde hispanique et luso-brésilien*, 110, 65-86. <https://doi.org/10.4000/caravelle.2897>

Nake, F. (2005). *Das doppelte Bild*. En M. Pratschke (Ed.), *Digitale Form: Vol. 3.2* (pp. 40-50). Berlin : Akad.-Verl.

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3), 337-343. <https://doi.org/10.1109/TIT.1977.1055714>