Compresión de datos utilizando la teoría de teselas

Data compression using theory of tiles

José Luis Hernández Hernández *
Feliciano Morales Severino
René Edmundo Cuevas Valencia ***

Fecha de recepción: 15 de abril del 2011
Fecha de aceptación: 16 de junio del 2011

Resumen

Esta investigación tiene como objetivo presentar un software que permita realizar la compresión de datos de un archivo, los cuales contengan cualquier tipo de información y que conjugando la teoría de las teselas[1] con la teoría de compresión de datos, se pueda reducir el espacio de almacenamiento de un archivo de datos no importando el equipo de cómputo utilizado, su plataforma ni el sistema operativo instalado en dicha computadora. Según la teoría de teselas, una tesela es una regularidad o patrón de figuras que cubre o pavimenta completamente una superficie plana y tiene la característica de la auto similitud; si este concepto se aplica a un archivo de datos, encontramos ciertas cantidades de datos, que son autosimilares o teselas y se repiten varias veces durante todo el archivo; de forma que con una representación que se almacene y las demás sean referenciadas, en cierto momento se puede generar el archivo original sin perder ni un solo dato, con el consecuente ahorro de espacio en el disco duro.

Palabras clave: compactación de datos, teoría de teselas, compresión de datos, compresión de datos con teselas.

* Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero, Av. Lazaro Cárdenas s/n, Ciudad Universitaria. Chilpancingo Guerrero México, Teléfono 01 (747) 47 2-79-43. Correo electrónico: tec_jlhh05@yahoo.com.mx
** Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero, Av. Lázaro Cárdenas s/n, Ciudad Universitaria. Chilpancingo, Guerrero México. Teléfono: 01 (747) 47 2-79-43. Correo electrónico: reneecuevas@hotmail.com.
*** Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero, Av. Lázaro Cárdenas s/n, Ciudad Universitaria. Chilpancingo Guerrero México. Teléfono: 01 (747) 47 2-79-43. Correo electrónico: sevefelici@hotmail.com

1 http://sites.google.com/site/tesela/, http://teselas4eso.wordpress.com/teselas/, http://divulgamat2.ehu.es/divulgamat15/index.php?option=com_content&view=article&id=11478&directory=67

Introducción

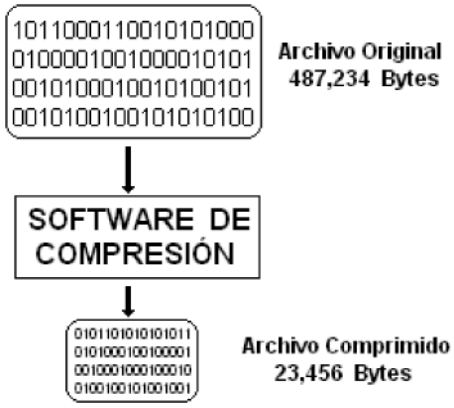
Es impresionante el avance tecnológico que ha tenido el procesamiento de datos en el periodo comprendido de 1985 a la fecha, tanto en equipo de cómputo como en productos de software. Los discos duros son muy importantes como medio de almacenamiento en los sistemas computacionales, porque pueden almacenar más datos y estos se pueden recuperar rápidamente (Ingelek, 1985).
En la actualidad, existen discos duros de tecnologías que permiten un tiempo de acceso muy corto; entre los líderes de este tipo de tecnologías se cuenta con empresas como Quantum, Seagate y Maxtor, las cuales fabrican discos duros con superficies aprovechables (libres de errores) en casi un 100% (Hernández, 2010).

Si se toma como referencia mayo del 2011, se da cuenta que existen una infinidad de capacidades de discos duros que van desde 350 gigabytes pasando por 500 Gigabytes y hasta llegar a discos de varios terabytes. Como se puede observar, los dispositivos de almacenamiento de datos han avanzado en forma vertiginosa, pero también el software es cada vez más complejo y los archivos de datos requieren cada vez más espacio.

En el caso de los archivos ejecutables y cualquier archivo que almacene una imagen, un sonido o un video, requieren una gran cantidad de espacio en bytes para poder almacenar completamente la información de la misma.

Durante muchos años, los usuarios de computadoras han discutido sobre las bondades y limitaciones de las tarjetas de video, tarjetas de audio, la memoria o los discos duros. Sin embargo, todos estos componentes llamados hardware habrían tardado muchos años más en evolucionar si no hubiera existido la necesidad de romper los límites físicos existentes a la hora de manejar la información. Desde luego, que se trata de la compresión de datos (ver la figura 1).

Figura 1. Proceso de compresión de un archivo



Una aplicación mal programada funcionará lentamente incluso en una máquina cuyo procesador sea de los de mayor velocidad. De hecho, si hay un elemento imprescindible a la hora de utilizar una computadora, ese es el software que se ejecuta. Siempre existirá un software para funcionar con cualquier tipo de hardware, mientras que un hardware sin el soporte software, es una computadora muerta (Pressman, 1996).

Algo parecido ha ocurrido con la compresión de datos: pocos usuarios le han dado la importancia que se merece, a pesar de ser uno de los más importantes logros informáticos ocurridos en los últimos veinte años.

La compresión no es más que una técnica que consiste en aplicar un algoritmo, es decir, una serie de transformaciones que reducen el tamaño inicial de un conjunto de datos informáticos, ya sean textos, archivos, gráficos, video o audio. Puesto que cualquier in-

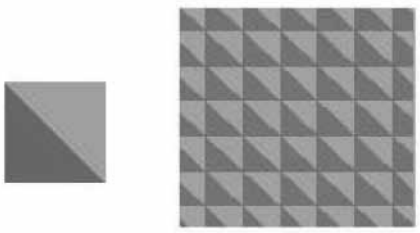
formación procesada por una computadora ocupa un determinado espacio a la hora de almacenarla, hay que encontrar la manera de reducir ese espacio, sin perder información. Esta es la base sobre la que se asientan todas las herramientas de compresión de datos.

Los logros obtenidos por la compresión de datos son espectaculares. Sin la existencia de compresores, es posible que el hardware de las computadoras hubiera evolucionado de forma muy distinta. La compresión de datos es un tema complejo, pero a la vez apasionante, pues no en vano abarca prácticamente todos los campos en los que se produce una manipulación de datos.

El funcionamiento de los compresores de archivos hace uso de una teoría muy sencilla. La idea consiste en partir de un determinado conjunto de archivos con un determinado tamaño y conseguir almacenarlos en un espacio mucho menor, sin perder información. Todos los compresores se aprovechan de la repetición de los datos dentro de un archivo, puesto que todos los archivos se traducen en combinaciones de bits, según el código ASCII, de forma que se aprecia que en la práctica existen secuencias de bytes repetidas, que pueden almacenarse de forma más corta. Por ejemplo, si un archivo contiene 300 bytes con el número 46, el cual corresponde al color de una determinada porción de un dibujo, estos 200 bytes se pueden reducir a 4 bytes, que contendrán el número 46 y el 200, para indicar que el primero está repetido 200 veces. Este método también se aplica a conjuntos de datos repetidos que no están seguidos, de manera que, dependiendo del sistema que se utilice para controlar los bloques repetidos, se obtienen los distintos algoritmos para la compresión de archivos.

Los bloques de bytes que se encuentran en forma adyacente y que generan un patrón, se pueden representar como teselas. Según la Real Academia Española, edición XIX del diccionario, la palabra tesela (del latín, *tessella*) significa "Cada una de las piezas cúbicas de mármol, piedra, barro cocido o cualquier otra material, con que los antiguos formaban los pavimentos de mosaico" (García Bellido, 1979).
Un patrón (Sánchez, 2002) o motivo (pattern) es una imagen que, colocada junto a copias de sí misma puede repetirse hasta el infinito sin que el dibujo así tenga rupturas. Los patrones o motivos son de varios tipos. El más sencillo es el que tiene como elemento básico (esa imagen que se repite), una loseta o tesela oblonga. Las más de las veces esta loseta es cuadrada, pero no es imprescindible y puede ser rectangular (ver la figura 2).

Figura 2. Ejemplos de patrones (izquierda patrón básico y derecha patrón de patrones)

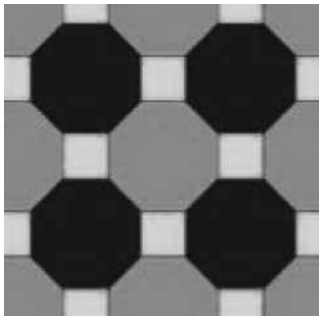


Los patrones encontrados en un archivo se pueden llamar teselas; aunque este término fue utilizado para referirse a las piezas con las que se cubría un piso, también se utiliza para referenciar con los patrones de bytes encontrados en un archivo (ver la figura 2).
En la figura 3, se encuentran tres teselas diferentes (la verde, la azul marino y la morada) y se encuentran perfectamente acomodadas en la imagen. En un archivo también se ubican bloques de bytes que se toman como patrones, por ejemplo, en un archivo de texto se encuentra la palabra "la" y la palabra "de" en varias partes de todo el texto; dichas palabras se pueden considerar como teselas.

Los patrones o teselas en un archivo de datos de cualquier tipo, contendrá varios patrones repetidos y que se encuentran al azar, pero que se pueden identificar claramente.

Proceso de compresión

Figura 3. Ejemplo de teselas



Fuente: http://reocities.com/SiliconValley/vista/2212/tesela.html

Para la elaboración del Software de Compresión de archivos utilizando teselas, se requiere analizar todo el archivo que se quiere comprimir y buscar patrones repetitivos, que serán considerados como teselas (entidades autosimilares) y dichas teselas serán representadas como un valor numérico entero (González, 1990) único, que deberá ser almacenado en un archivo de teselas para utilizarlo posteriormente en la descompresión.
Para esto, es necesario manejar un archivo para almacenar las teselas obtenidas, que algunos autores suelen llamar diccionario, el cual, al principio, estará vacío y conforme se usa el compresor se almacenarán las teselas, que serán patrones que posiblemente se repitan más adelante en el mismo archivo o bien en otros archivos que se compriman posteriormente.

Las nuevas teselas que se van generando se van almacenando en un archivo, debido a que cualquier tesela conocida ya no se almacena; simplemente se utiliza para la compresión y la descompresión de un archivo. Cabe hacer mención que al momento de encontrar nuevas teselas siempre se verifica la tesela obtenida, contra las que ya existen y si existe se toma su valor numérico entero para representarlo en el archivo que se está comprimiendo.

Desafortunadamente, las teselas se van generando en una forma estática se define como los datos necesarios para representar la tesela o patrón. Dichos datos se representan mediante la estructura que se muestra en la figura 4.

Figura 4. Estructura para almacenar las teselas que se van obteniendo

TESELA	ELEMENTO IZQUIERDO	ELEMENTO DERECHO
UN NÚMERO	UN NÚMERO	UN NÚMERO

Fuente: elaboración propia.

La estructura de la figura 3 es utilizada y se almacena en un archivo que es llamado MAESTRO.TES y se encontrará en la unidad C en la carpeta C:\WINDOWS de la computadora en uso. Con esta forma de compresión, los patrones que ya se convirtieron en teselas simplemente se utilizan; el inconveniente es que cada vez que se genera una nueva tesela se acumula en el archivo maestro de teselas que sigue creciendo.

Conversión de teselas a números enteros

El proceso de teselado es un mecanismo que permite la representación de todas las características de un objeto, figura, señal, etc.; por un elemento numérico que contiene referencia de todos y cada uno de los elementos representados. Esa idea parte del hecho de que

existe un elemento o grupo de elementos que tiene las propiedades básicas de autosimilitud y que por medio de ello es posible reproducir todas y cada una de las posibles combinaciones que pueden darse con todo el código ASCII.

En el caso de un archivo de cualquier formato, los bytes que están almacenados uno después del otro corresponden al código ASCII, este grupo de elementos bien pueden ser: las letras, los números, los caracteres especiales y los caracteres de control, de forma que cada byte tiene un valor comprendido entre el rango de 0 a 255 según el código ASCII, de este modo, es posible simplificar un bloque de bytes y representarlos mediante un solo número (Ingelek, 1895).

Precisamente esta forma de representación de autosimilitud tiene mucho que ver con el concepto de teselas del cual se ha hecho mención anteriormente. Esta propiedad permite reconocer algún elemento que tenga las características del todo, del cual se extrae y que permita al replicarse regenerar ese todo bajo la guía de una función o estructura determinada.

La aplicación de este concepto al campo de la compresión de archivos se inicia con dos bytes adyacentes de un archivo y se reemplazan estos por alguna forma de representación de sus elementos originales, en este caso, podría corresponder a cada par de bytes un número entero, por lo que se da una representación a cada par de elementos encontrados para generar una nueva representación equivalente. Este proceso se repite hasta encontrar una representación única para un conjunto de *n* bytes. Entonces, se encuentra lo qur representa en forma condensada a un bloque de bytes. Con esto, el proceso de almacenamiento es muy eficiente, ya que en lugar de almacenar todo el bloque de bytes original se puede almacenar solamente su representación. Para recuperar dicho bloque, solamente es necesario seguir el proceso inverso, es decir, de sustitución directa de cada nivel de representación encontrado. Para lograrlo se van almacenando cada uno de los pares de elementos representados.

Al realizar múltiples operaciones sobre un bloque de bytes originales, aparecerán cada vez con mayor frecuencia pares representados que ya han aparecido por lo que, en un momento dado, dichos pares existen y mediante ellos será posible representar cualquiera de los elementos del bloque de bytes original; esto implica que habremos encontrado las teselas para dicho bloque de bytes. Es claro que para este caso, las teselas más elementales son los bytes que conforman al código ASCII, ya que con ellos es posible formar cualquier palabra, número o dato. Aún así, es necesario encontrar una forma más compacta de representar este bloque de bytes y esto se logra encontrando alguna representación que tenga teselas con mayor grado de replicación sobre cualquier bloque de bytes a representar.

Caso práctico de compresión con teselas

Procedimiento de la compresión utilizando teselas con valores enteros.
Texto inicial = **"Compresión"**

Su representación numérica de acuerdo con el código ASCII es:

67	111	109	112	114	101	115	105	162	110

De donde se registran y se almacenan los siguientes valores:

C → 67	e → 101
o → 111	s → 115
m → 109	i → 105
p → 112	ó → 162
r → 114	n → 110

Ahora del texto original, se toman pares de elementos y se sustituyen por lo que serán los primeros pares de teselas. Para este caso se propone tomar pares, pero podrían ser tomadas tercias o cada número representativo del texto original con un símbolo de unión entre caracteres. De esta forma, se inicia la identificación de pares desde el número 256 que corresponde a un valor después del último valor del código ASCII, por lo que se tiene que:

67	111	109	112	114	101	115	105	162	110
256		257		258		259		260	

Se caracteriza ahora como:

256	257	258	259	260

Si a dicha representación le aplicamos el mismo mecanismo, tenemos que:

256	257	258	259	260	-1
261		262		263	

Cabe hacer mención que el valor -1 es un elemento vacío y se utiliza exclusivamente para poder tener un par de elementos y crear la tesela. Si dentro de los pares que se van almacenando se presenta un par que ya existe entonces no se almacena. Ahora el texto se representa de la forma siguiente:

261	262	263

Si a lo que se obtuvo se le aplica el mismo proceso tenemos que:

261	262	263	-1
264		265	

Y obtenemos la siguiente representación:

264	265

Si a lo que obtuvimos le aplicamos el mismo proceso tenemos que:

264	265
266	

De esta forma se tiene que el número 266 representa al texto "Compresión". Esto se puede representar en forma más clara de la forma siguiente:

C	o	m	p	r	e	s	i	ó	n	
67	111	109	112	114	101	115	105	162	110	
256		257		258		259		260		-1
261				262				263		-1
264								265		
266										

Durante el proceso se generaron once teselas, que serán almacenadas en el archivo maestro de teselas. Dichas teselas generadas se muestran en la tabla 1.

Tabla 1. Teselas generadas en la compresión

Tabla de teselas	
256 → 67, 111	262 → 258, 259
257 → 109, 112	263 → 260,-1
258 → 114, 101	264 → 261,262
259 → 115, 105	265 → 263, -1
260 → 162, 110	266 → 264,265
261 → 256, 257	

Fuente: elaboración propia

Resultados de la compresión de archivos

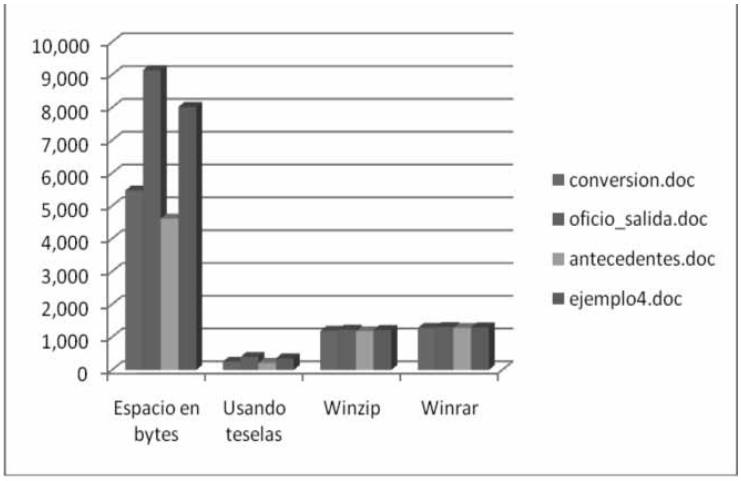
Se tomaron cuatro archivos de Word que tienen extensión .DOC y se procedió a comprimirlos con el compresor de archivos propuesto, con el Winzip y con el Winrar obteniéndose los resultados mostrados en la tabla 2.

Tabla 2. Cantidad de bytes requeridos sin compresión y comprimidos con 3 compresores diferentes

Archivo	Espacio en bytes	Usando Teselas	Winzip	Winrar
conversion.doc	5,471	253	1,192	1,285
oficio_salida.doc	9,142	397	1,229	1,305
antecedentes.doc	4,615	219	1,182	1,281
ejemplo4.doc	8,021	355	1,217	1,298

Con los datos obtenidos de los cuatro archivos comprimidos, se generó una gráfica comparativa y se obtuvieron los resultados mostrados en la figura 5.

Figura 5. Cantidad de bytes requeridos sin compresión y comprimidos con tres compresores diferentes



Fuente: 148.204.103.95/somece2009memorias/.../HernandezHernandezJoseLuis.do...

Agradecimientos

Durante el año 2010 y lo que va del 2011, el cuerpo académico de Tecnologías de la Información y Comunicaciones de la Unidad académica de Ingeniería de la Universidad Autónoma de Guerrero, ha recibido apoyo del M. en C. Juan Carlos Medina Martínez, presidente drrrrrrel consejo de Unidad y director de la Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero.

Conclusión

De conformidad con el objetivo propuesto en este trabajo e investigación, se consultó la teoría de: compresión de archivos, teoría de teselas, ingeniería de software y modelos orientados a objetos.

En cuanto al beneficio se puede utilizar el compresor elaborado con una certidumbre del 99,8 %. Se utilizó en forma personal para comprimir aproximadamente cincuenta archivos y funcionó correctamente

al 100%, tanto en su compresión como en su descompresión.

Mediante el desarrollo e implementación de este compresor de archivos, fueron surgiendo cada vez nuevos enfoques, extensiones y mejoras a este. Como proyecto de investigación tiene sus limitaciones y alcances, y para mejorarlo se pueden proponer otros trabajos complementarios.

Referencias

Ediciones Ingelek (1985). *Enciclopedia práctica de la informática. Cintas y discos* (1ª ed.).. Ediciones Nueva Lente.

García Bellido, A. (1979). Arte romano. *Enciclopedia clásica C.S.I.C.* Madrid.

González Mari, J.L. (1990). *Los números enteros*. Síntesis,.

Hernández Hernández, J.L. (2010). *Administración de archivos. Ejemplos prácticos de utilización de archivos*: secuenciales, indexados y directos, utilizando el lenguaje C++". E-book.

Pressman, R.S. *Ingeniería del Software. Un enfoque práctico* (3ª ed.). Mc. Graw Hill.

Sánchez, G. (2002). *Introducción a los motivos y patrones*. Recuperado de: http://www.gusgsm.com/motivos_repetitivoshttp://www.gusgsm.com/motivos_repetitivos .

http://sites.google.com/site/tesela/

http://teselas4eso.wordpress.com/teselas/ , http://divulgamat2.ehu.es/divulgamat15/index.php?option=com_content&view=article&id=11478&directory=67