

UN ALGORITMO DE MARCADO REVERSIBLE PARA IMÁGENES FIJAS BASADO EN RUN-LENGTH

Laura M. Vargas[†], Elizabeth Vera[‡]

[†]Departamento de Computación, Laboratorio de Redes y Comunicaciones de Datos (LaRyC), Facultad de Ciencias Exactas, Físicas y Naturales - Universidad Nacional de Córdoba, Córdoba, Argentina, lvargas@efn.uncor.edu

[‡] Departamento de Matemática, Laboratorio de Procesamiento de Señales (LaPSe), Facultad de Ciencias Exactas, Físicas y Naturales- Universidad Nacional de Córdoba, Córdoba, Argentina, epayer@efn.uncor.edu

Resumen: En este trabajo se aborda un algoritmo de marcado reversible para imágenes fijas. Este se apoya en las secuencias de valores iguales que se presenten en una imagen, lo que lo hace aplicable en especial a imágenes médicas. Es un marcado reversible, por lo tanto frágil, de detección ciega (sin necesidad de conocer la imagen original) que utiliza los LSBs (Bits Menos Significativos) preservando la calidad de la imagen con poca complejidad computacional. Mediante este método se puede autenticar la imagen así como embeber metadatos o resúmenes que permiten asegurar la integridad de la misma.

Palabras Clave: *marcas de agua, imágenes fijas, algoritmos, seguridad informática, RL (Run Length)*

1. INTRODUCCIÓN

Las técnicas de marcado de agua o watermarking se han desarrollado fuertemente en las dos últimas décadas con el objetivo de colaborar en la resolución de problemas asociados con la autenticación o aseguramiento de procedencia e integridad de archivos digitales, imágenes, audio y video digital, ya que estos pueden sufrir alteraciones tanto en el almacenamiento a largo plazo como en la transmisión de los mismos, siendo uno de sus precursores Cox [2]. Mediante distintas variantes, los métodos de marcado introducen en el objeto anfitrión información que es capaz de realizar importantes contribuciones en el sentido indicado. En imágenes, objeto de este trabajo, el marcado invisible se puede realizar en el dominio espacial, en cuyo caso se suele utilizar el bit menos significativo (Least Significant Bit/LSB) de los píxeles para esconder la marca, o bien se recurre a la utilización de transformadas matemáticas y se inserta la marca en el dominio de la transformada. Está particularmente extendido el uso de la transformada discreta del coseno (DCT), así como el de la transformada discreta wavelet (DWT).

Los algoritmos de marcado pueden ser *reversibles* o *irreversibles*. En los reversibles, las marcas siempre son frágiles, es decir que se destruyen ante ataques, intencionales o no, siendo los preferidos para preservar imágenes legales [3].

Las marcas de agua pueden considerarse señales con información. La cantidad de información extra o *payload* que permiten incorporar al objeto anfitrión o *host* se conoce como *capacidad* de la marca y, en el caso de imágenes, se evalúa en bpp (bits por píxel) según (1), donde N_{bm} es la cantidad de bits de la marca.

$$Capacidad(bpp) = \frac{N_{bm}}{Cantidad\ total\ pixeles\ imagen} \quad (1)$$

Para evaluar la calidad de la imagen marcada se utilizan los parámetros MSE (Mean Square Error) y PSNR (Peak Signal Noise Ratio) indicados en (2) y (3), respectivamente, donde $I(i,j)$ es el píxel de la imagen original, $I_w(i,j)$ el píxel de la imagen marcada, M es la cantidad de filas y N la de columnas de la matriz imagen e I_{pico} es su máxima intensidad. También se emplea el parámetro SSIM (Structural Similarity Index), según (4). En este se consideran valores medios μ y varianzas σ de la imagen original I y la marcada I_w , así como coeficientes c_1 y c_2 que dependen del rango dinámico de intensidad [5].

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (I_w(i,j) - I(i,j))^2}{MN} \quad (2)$$

$$PSNR(dB) = 10 \log_{10} \frac{[I_{pico}]^2}{MSE} \quad (3)$$

$$SSIM = \frac{(2\mu_I \mu_{I_w} + c_1)(2\sigma_{I_w} + c_2)}{(\mu_I^2 + \mu_{I_w}^2 + c_1)(\sigma_I^2 + \sigma_{I_w}^2 + c_2)} \quad (4)$$

En este trabajo se presenta un algoritmo reversible (por lo tanto, frágil) para imágenes fijas que, al permitir recuperar la imagen original, puede emplearse en imágenes médicas, forenses o cartográficas. En el punto 2, se presenta el algoritmo a desarrollar, en el punto 3, resultados de la experimentación efectuada y en el punto 4, conclusiones y futuros trabajos.

2. ALGORITMO BASADO EN RL (RUN-LENGTH)

2.1 INTRODUCCIÓN

Las secuencias de valores repetidos que se presentan en imágenes se conocen como RL (Run Length) son aprovechadas por métodos de compresión como Run Length Encoding (RLE). Estas también se pueden utilizar en estrategias de embebido [1]. Sea una imagen I de M filas y N columnas. Conviene convertir la matriz inicial en un vector v de dimensión $L=M \times N$, para facilitar la programación. Si se presentan en dicho vector s cadenas de píxeles de igual valor de intensidad, se puede conservar el primer píxel que servirá luego en la restauración y modificar los valores de los siguientes según la marca, cambiando en estos k LSBs, variable que controlará la capacidad del algoritmo, influyendo también en la calidad de la imagen marcada.

De cada secuencia se precisa almacenar posición $p(s_i)$ y longitud o cantidad de elementos $l(s_i)$, para poder recuperar la imagen original. Para indicar la posición p de una secuencia se precisarán:

$$N_{p(s)} = \lceil \log_2 L \rceil \text{ bits} \quad (5)$$

Para indicar la longitud de una secuencia $l(s)$ en bits se puede tomar el valor necesario para la longitud máxima presentada en una secuencia l_{smax} , en cuyo caso (6) también debe ser informado.

$$N_{l(s)} = \lceil \log_2 l_{smax} \rceil \text{ bits} \quad (6)$$

Si se considera que cada secuencia debe almacenar la longitud propia y la posición de la siguiente, se plantea una restricción para que una secuencia embeba información. Así, para que una secuencia sea útil debe cumplir la condición (7).

$$l(s_i) > \left\lceil \frac{N_{p(s)}}{k} \right\rceil + \left\lceil \frac{N_{l(s)}}{k} \right\rceil + 1 \quad (7)$$

La totalidad de datos auxiliares que se deberán embeber se indican en la figura 1, donde CT indica la cantidad total de secuencias que se usarán, valor necesario para conocer el final de la cadena en el proceso de recuperación.

k	CT	$p(s_1)$	$l(s_1)$	$p(s_2)$	$l(s_2)$...
-----	----	----------	----------	----------	----------	-----

Figura 1 – Estructura de datos auxiliares

Wang propone embeber los datos auxiliares en una zona reservada donde, en lugar de utilizar el algoritmo general, se sobrescribe el bit menos significativo. En ese caso, el valor del bit original debe guardarse, agregándose a la marca, para poder tener la reversibilidad deseada. Otra opción es utilizar este procedimiento solo para guardar los campos k , CT y posición de la primera secuencia, mientras que en cada secuencia se almacena la longitud de la misma y la posición de la siguiente.

La ventaja más importante del algoritmo se obtiene haciendo que la marca no modifique el valor de intensidad original del píxel, sino un valor llamado *base*, así se evita el overflow o underflow (desbordamiento) con poca complejidad computacional, a diferencia de algoritmos clásicos como el de Tian [4]. En la implementación realizada se optó por determinar la *base* según (8). Su cálculo es el mismo en la secuencia de la imagen original (s_i) y la marcada (s'_i), ya que toma en cuenta el primer píxel de la secuencia y este se conserva sin modificaciones.

$$\begin{aligned} base(s_i) = base(s'_i) &= v(s_{(i,1)}) - 2^k - 1 & \text{si } v(s_{(i,1)}) \geq 2^k - 1 \\ base(s_i) = base(s'_i) &= 0 & \text{si } v(s_{(i,1)}) < 2^k - 1 \end{aligned} \quad (8)$$

El nuevo valor de intensidad v_w de los elementos de la secuencia será, según (9):

$$v_w(s_{i,j}) = base(s_i) + m \quad j > 1 \quad \text{y} \quad j \leq l(s_i) \quad (9)$$

donde i indica el número de secuencia, j la posición del elemento dentro de la secuencia, l_s la longitud de la misma y m un bit de marca. Para una imagen cuyo valor de intensidad en la secuencia i sea igual a 255 y con $k=1$, se tiene:

$$v(s_{(i,1)}) = 255; \quad 2^k - 1 = 1; \quad base(s_i) = 254 \quad \left. \begin{array}{l} v_w(s_{(i,j)}) = base(s_i) + 0 = 254 \quad \text{bit de marca} = 0 \\ v_w(s_{(i,j)}) = base(s_i) + 1 = 255 \quad \text{bit de marca} = 1 \end{array} \right\} \text{Para } 1 < j \leq l_{(s_i)}$$

2.2 EMBEBIDO

Se obtiene un vector a partir de la matriz imagen, para facilidad de manipulación computacional. Se fija k . Se recorre el vector y se determinan las secuencias útiles mediante la ecuación (7). Se obtiene CT. Se embeben los datos auxiliares en zona reservada. Se calcula el valor de intensidad base de las secuencias según (8) y el valor marcado según (9). Se convierte el vector en matriz.

2.3 RESTAURACIÓN

Se convierte la matriz imagen en vector. Se extraen los datos auxiliares de la zona reservada. De acuerdo a la información que brindan, se buscan los píxeles marcados y, previo cálculo del valor base según (8), se recuperan los elementos del vector marca mediante la ecuación (10).

$$m = v_w(s_{i,j}) - base(s_i) \quad j > 1 \quad y \quad j \leq l_{s(i)} \quad (10)$$

Una parte de la marca son los bits sobrescritos al embeber los datos auxiliares. Estos bits se utilizan para restaurar el valor original en la zona de almacenamiento de datos auxiliares. Finalmente, se restauran los valores de los píxeles de las cadenas utilizadas, utilizando el valor del primer píxel.

La capacidad teórica máxima del algoritmo, es decir la cantidad de información extra que se puede agregar, se indica en (11).

$$C(bits) = \sum_1^{CT} \left[(l_{(s_i)} - 1) \times k - N_{p(s)} - N_{l(s)} \right] - bits(k) - bits(CT) \quad (11)$$

3. EXPERIMENTACIÓN

Se empleó MatLab como lenguaje de programación. Se utilizaron secuencias binarias pseudoaleatorias para simular la marca. En imágenes comunes, el algoritmo resultó inaplicable porque no presentan secuencias de repeticiones suficientes. Se experimentó con imágenes médicas en formato Dicom con datos enteros sin signo de 8 y 16 bits y se comparó con el algoritmo de Tian. Los resultados se muestran en la tabla 1, las imágenes en las figuras 2 a 5.

Imagen (512x512 píxeles)	Capacidad (bpp)	k	PSNR (dB)	MSE	SSIM
'cr-intest' (8bpp)	0,17	1	56,68	0,096	0,9956
		2	52,12	0,273	0,9864
	0,30	2	48,87	0,570	0,9728
		3	43,77	1,868	0,9527
	0,50	3	41,70	3,005	0,9239
0,80	4	34,48	15,868	0,8671	
'cr-ri' (8bpp)	0,17	1	56,07	0,096	0,9957
		2	51,01	0,304	0,9848
	0,30	2	48,28	0,571	0,9729
		3	43,97	1,852	0,9522
	0,50	3	42,48	3,011	0,9237
0,80	4	34,68	15,718	0,8680	
'mr-cerebro' (16bpp)	0,17	1	75,73	0,123	0,9938
		2	71,20	0,358	0,9804
	0,30	2	68,59	0,639	0,9631
		3	63,78	1,832	0,9413
	0,50	3	61,65	3,165	0,9023
0,80	4	54,50	16,410	0,8372	
'mr-espina' (16bpp)	0,06	1	78,69	0,029	0,9992
	0,10	2	71,08	0,173	0,9947
	0,16	3	63,89	0,901	0,9854

Tabla 1- Evaluación del algoritmo: capacidad y calidad

Se variaron k y la carga útil. Se observa que, para igual capacidad, la calidad es superior para el menor valor de k . En imágenes con menos secuencias útiles, para tener valores similares de capacidad, se precisa aumentar k a costa de la calidad. Capacidad y calidad dependen de la imagen en particular. Si los píxeles se representan con 16 bits y hay secuencias suficientes, se alcanza mayor capacidad, manteniendo una buena calidad. Una imagen como la de la figura 5, con pocas repeticiones, tiene muy poca capacidad.



Figura 2- 'cr-intestino'



Figura 3- 'cr-ri'



Figura 4- 'mr-cerebro'



Figura 5 - 'mr-espina'

Algoritmo	cr-intestino		cr-ri		mr-cerebro		mr-espina	
	Carga útil (bpp)	PSNR(dB)						
Tian	0,44	40,41	0,47	42,14	0,33	57,29	0,24	46,03
RL	0,50	41,74	0,50	42,48	0,50	61,65	0,16	63,89

Tabla 2- Comparación del algoritmo RL con el de Tian

En la tabla 2, se comparan los valores obtenidos con los arrojados por el algoritmo reversible de Tian. Este último se vale de la transformada entera, utiliza diferencias entre pares de píxeles adyacentes para insertar información. Por ello, tiene un límite teórico de capacidad de 0,5 bpp. Para superar este valor es necesario repetir el algoritmo, por ejemplo primero se recorre la imagen en sentido horizontal y luego en el vertical. Otra variante de Tian es trabajar con bloques, así si se toma una cuaterna se llega a un valor teórico máximo de 0,75 bpp. Para efectuar la comparación se determinó la máxima capacidad para cada imagen según Tian simple y luego se trató de obtener un resultado similar, o mejor, con Run-Length. En el caso de la imagen de la Figura 5, Tian supera a RL.

4. CONCLUSIONES Y FUTUROS TRABAJOS

Este algoritmo RL presenta ventajas respecto a otros porque resuelve con bajo costo computacional el desbordamiento y no necesita embeber el mapa de ubicación de las marcas, como el de Tian y los basados en él que deben comprimirlo para preservar la capacidad. Se buscó mejorar capacidad y calidad, variando k y el embebido de datos auxiliares. El método RL mantiene una buena calidad de la imagen marcada, para valores bajos de k . Se puede predefinir k , para cada tipo de imagen médica. Su desventaja es que solamente es utilizable en imágenes con cadenas de valores de intensidad iguales. Para preservar la marca se deben usar métodos de corrección de errores. El orden de inserción debe depender de una clave, lo que se implementará en futuros trabajos. También se está trabajando en automatización de variantes de Tian.

La marca imperceptible que se inserta es útil para introducir metainformación (datos sobre el origen del archivo, datos del paciente en el caso de imágenes médicas) la que puede estar encriptada, o bien para insertar un resumen o digesto como el SHA o MD5 que permita decidir luego si la imagen fue adulterada.

REFERENCIAS

- [1] A. CHANG, J. XU, *Reversible Run Length Data Embedding for Medical Images*. En Communications Software and Networks (ICCSN), IEEE 3th International Conference, Conference Publications (2011), pp. 260-263.
- [2] I. COX, J. KILIAN, F. LEIGHTON, T. SHAMOON.: *Secure Spread Spectrum watermarking for images, audio and video*. IEEE Transactions on Image Processing Vol. 6 (1997), pp. 1673-1687.
- [3] G. COATRIEUX, H. MAITRE, B. SANKUR, Y. ROLLAND, R. COLLOREC, *Relevance of Watermarking in Medical Imaging-* Proceedings of the IEEE EMBS Conf. on Information Technology Applications in Biomedicine, Arlington, USA (2000), pp. 250-255.
- [4] J. TIAN, *Reversible data embedding using a difference expansion*. En IEEE Transactions on Circuits and Systems for Video, Vol.13, N°8 (2003), pp. 890-896.
- [5] Z. WANG, A. BOVIK, H. SHEIKH Y E. SIMONCELLI, *Image quality assessment: From error visibility to structural similarity*. En IEEE Transactions on Image Processing, Vol. 13, N°4 (2004), pp. 600-612.