INVESTIGACIÓN & DESARROLLO, No. **11**, Vol. **1**: 137 – 148 (2011)
ISSN 2518-4431

WATERMARK RESISTENTE EN EL DOMINIO DE LAS FRECUENCIAS DE IMÁGENES DIGITALES PARA SU AUTENTICACIÓN SEGURA MEDIANTE AUTÓMATAS CELULARES

A RESISTANT *WATERMARKING* IN THE FREQUENCY DOMAIN OF DIGITAL IMAGES FOR SECURE AUTHENTICATION THROUGH CELLULAR AUTOMATA

Luz Fátima Huallpa Vargas*, Luis Pánfilo Yapu Quispe**
**Academia Local de Networking Cisco – Ingeniería Informática*
Universidad Autónoma Tomás Frías, Bolivia
luz.fatima@ciscouatf.edu.bo
***Investigación - Ingeniería Informática*
Universidad Autónoma Tomás Frías, Bolivia
luis.yapu@gmail.com
(Recibido el 01 de octubre 2011, aceptado para publicación el 03 de diciembre 2011)

RESUMEN

En este trabajo se presenta el desarrollo de un procedimiento de autenticación de imágenes digitales combinando diversos métodos relacionados con *watermarks* (marcas de agua) digitales y métodos de cifrado, con el fin de proteger los derechos de autor de imágenes digitales. Para que este procedimiento sea robusto, imperceptible y resistente a diferentes ataques intencionales o no intencionales se trabaja en el dominio de las frecuencias de la imagen utilizando la Transformada Discreta del Coseno (*Discrete Cosine Transform*, DCT). Para agregar mayor nivel de protección y seguridad a la imagen marcada, se cifra el mensaje insertado a partir de una clave del autor empleando el método de Autómatas Celulares. El mensaje insertado es resistente a la compresión JPEG, a los ruidos y filtros gaussiano y *salt-and-pepper*. La extracción del mensaje se realiza a partir de la imagen marcada para lo cual se hace uso estricto de la clave del autor para verificar la autenticidad del mismo.

ABSTRACT

This work presents an authentication procedure for copyright protection of digital images that combines different methods of digital watermarking and encryption. In order to ensure that the authentication procedure is robust, imperceptible, and resistant to various intentional and unintentional attacks, the proposed approach works in the frequency domain of digital images and uses the *Discrete Cosine Transform* (DCT). To add a higher level of protection and security to the watermarked digital image, the message to be inserted is encrypted using the author's password and the Cellular Automata method. The inserted message is resistant to the JPEG compression and to Gaussian and *Salt-and-Pepper* noises and filters. The message can be extracted from the watermarked image by the strictly usage of the author's password, so as to verify the authenticity.

Palabras Clave: *Watermark Digital*, Transformada Discreta del Coseno, Autómatas Celulares, Ruido Gaussiano, Ruido *Salt-and-Pepper*.
Keywords: Digital watermarking, Discrete Cosine Transform, Cellular Automata, Gaussian Noise, Salt-and-Pepper Noise.

1. INTRODUCCIÓN

El creciente avance tecnológico de la sociedad permite intercambiar cualquier tipo de información en formato digital (textos, imágenes, video, audio, software) en la red mundial Internet. La facilidad de modificación y uso múltiple de la información digital va en contra de los derechos de autor, es en ese sentido que se ha expandido el concepto de *watermark* digital para verificar la propiedad de una información. Algunas técnicas de *watermarking* han sido propuestas con el objetivo de proteger los derechos de autor de documentos digitales, de manera que se logre insertar un mensaje o firma digital que identifique a un usuario como autor del documento [1].

Un *watermark* es un mensaje oculto insertado en otro mensaje portador. El *watermark* puede contener información acerca del autor del documento u otra que permite verificar la autenticidad de éste. En el *watermarking* multimedia el mensaje portador es una imagen, audio o video.

El *watermark* digital debe cumplir con las siguientes características, Cedillo *et al*. [2]:

- Debe ser imperceptible, es decir no ser visible al Sistema Visual Humano (SVH).

DOI: 10.23881/idupbo.011.1-9i

- Resistente a diferentes ataques, sean estos intencionales lo que se refiere a manipulaciones sobre la imagen con el fin de eliminar la protección de derechos de autor tales como deformaciones geométricas, rotación, escalamiento entre otros, o no intencionales como es el caso de la compresión JPEG y otros ruidos que podrían añadirse.
- No ambiguo, con el objetivo de poder recuperar el mensaje empotrado en la imagen digital. La no ambigüedad en el proceso de detección significa que el mensaje extraído debe ser lo suficientemente claro para poder demostrar la propiedad del autor.

La forma de adquirir el material de multimedia resulta muy sencilla y es ahí donde surgen los numerosos ataques y amenazas, por lo que el copiar, modificar, reproducir, hasta obtener copias exactas sin autorización pueden vulnerar los derechos de propiedad, sobre todo intelectuales.

Una referencia muy actual del área es el libro [1], sobre todo los capítulos 9, 10 y 11 explican métodos estándares para resolver los problemas de robustez, seguridad y autenticación. Otra referencia es el curso avanzado de *Multimedia Security* de Voloshynovskiy, Koval y Pun [3][1] de la Universidad de Ginebra. En estas notas de curso se explican, en particular, nociones y técnicas básicas de criptografía y teoría de la información que pueden incluirse para mejorar el nivel de seguridad. Una buena referencia general en criptografía es el libro de Stinson [4] y un libro clásico en teoría de la información es Cover & Thomas [5].

Existen varios métodos que fueron propuestos para tratar de resolver algunos de los problemas, considerando las características que debe tener un watermark. Algunos utilizan propiedades matemáticas de transformadas en dominios invariantes como es la Transformada Fourier-Mellin (FMT) [6], pero que necesita una interpolación de píxeles que llega a provocar una cierta distorsión de la imagen. Otros métodos como [7] y [8] están basados en el histograma de la imagen, son robustos a algunas deformaciones geométricas pero no a compresión JPG, ruidos añadidos o filtros usuales. En [9] se inserta, además del watermark, un *template* llamado *pilot* para estimar el nivel de deformación y que posteriormente se utiliza para reconstruir la imagen antes del proceso de detección. Un problema potencial en este método es que sería posible la detección del template si se obtienen varias imágenes marcadas. El método propuesto en [10] utiliza la obtención de puntos característicos en la imagen que permiten definir triángulos dentro de los cuales se inserta el watermark. Este método requiere un cálculo robusto de los puntos especiales y utiliza interpolación mediante *splines* cúbicos lo que puede causar distorsiones en espacios discretos como son las imágenes digitales. Finalmente, una familia métodos fue iniciado por Hu en [11] mediante la teoría de momentos invariantes. Dentro de éstos métodos, el trabajo de Dong *et al.* [12] es una propuesta robusta a distorsiones geométricas que normaliza un patrón y realiza la inserción en el dominio espacial para disminuir las distorsiones. Para reducir la tasa de bits erróneos, [2] presenta una variante en la que el watermark se inserta sólo en ciertas regiones de la imagen que fueron reconocidas como de textura fuerte [13], es decir zonas susceptibles de ser modificadas pero con que no son fácilmente reconocibles por el SVH.

Este trabajo retoma la propuesta [2] añadiendo las siguientes características:

- Se presenta una forma matricial de cálculo de la DCT que permite observar esta transformada como una conjugación por una matriz ortogonal. Esto permite fácilmente obtener la transformada inversa invirtiendo la ecuación y para así poder utilizar la misma rutina en el cómputo de ambas.
- Se cifra el mensaje a ser insertado utilizando el método de autómatas celulares. Esto añade el nivel de seguridad sin retardar demasiado el proceso de inserción y extracción.
- Se realiza una modificación a la Ecuación (13) de la referencia [2] con el fin de evitar una saturación posible al añadir un watermark a una imagen que contiene pixeles con valores próximos al límite permitido que es 255. La ecuación utilizada en este trabajo es (7).
- El prototipo se realizó en C#, cuyo entorno *user-friendly* permite al usuario marcar sus propias imágenes simplemente siguiendo una secuencia de botones.

El resto del artículo está organizado como sigue: en la Sección 2 de este trabajo se presenta la descripción de los procesos que se emplean para autenticar una imagen: inserción y extracción, en la Sección 3 se resumen los resultados de las pruebas realizadas y finalmente en la Sección 4 se presentan las conclusiones del trabajo.

2. DESCRIPCIÓN DE LOS PROCESOS PARA LA AUTENTICACIÓN DE IMÁGENES

Cedillo Hernández *et al*. [2] mencionan que para la inserción-extracción del *watermark* de manera robusta e imperceptible, se deben seguir tres procesos que son: 1) Proceso de normalización de la imagen, 2) Proceso de inserción del watermark y 3) Proceso de detección-extracción del watermark.

[1] Disponible bajo solicitud a los autores.

En el proceso de normalización de la imagen se emplea la teoría de momentos invariantes [11]. A continuación se va a describir el procedimiento de cálculo de momentos invariantes.

- **Momentos Invariantes**

Los **momentos geométricos** $m_{p,q}$ y los **momentos centrales** $\mu_{p,q}$ de la imagen *f(x,y)* se definen de la siguiente forma:

$$m_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} x^p y^q f(x,y) \qquad \mu_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{1}$$

donde $p, q = 1,2,\ldots$, toman valores enteros e indican en orden de los momentos, $f(x,y)$ denota la imagen digital de tamaño M x N, $(\bar{x},\bar{y})$ es el centro de masa de la imagen que se obtiene mediante las fórmulas:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

2.1. Procedimiento de normalización de imágenes

Para obtener una imagen normalizada se siguen los siguientes pasos:

(1) Obtener la imagen original que se desea proteger, a la que se denomina *f*(*x*,*y*), que mide la luminancia del pixel en la posición (*x*,*y*). En un gráfico en colores se tienen 3 funciones *f*(*x*,*y*), una por plano RGB (red, green, blue).
(2) Convertir en escala de grises la imagen *f*(*x*,*y*).
(3) Normalizar la imagen empleando la técnica propuesta por Dong *et al.* [12].

a. A partir de la imagen *f*(*x*,*y*).se procede a obtener la nueva imagen $f_1(x,y)$ realizando una translación con el vector (d_1,d_2).

Los valores d_1 y d_2 son los momentos geométricos normalizados $\bar{x}$, $\bar{y}$, cuyos valores indican el centro de masa de la distribución de grises de *f*(*x*,*y*), como se muestra en la Figura 1.

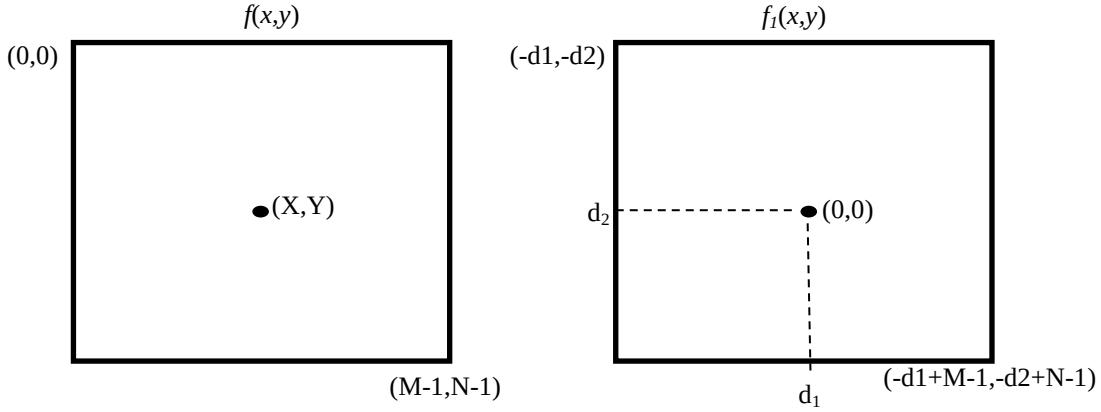

Figura 1- Ubicación gráfica del centro de masa de *f*(*x*,*y*) y $f_1(x,y)$.

b. Realizar una transformación de deformación en el eje *x* a la imagen $f_1(x,y)$. La matriz que permite realizar la deformación horizontal es A_x que actúa en el rectángulo como se muestra en la Figura 2.

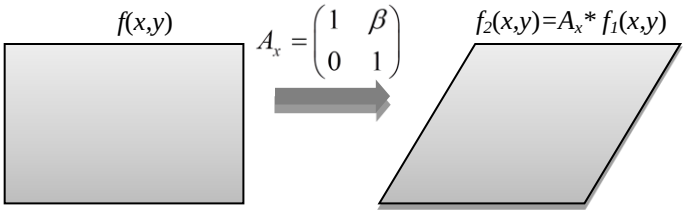

Figura 2: Deformación horizontal de *f*(*x*,*y*) mediante la matriz $A_{x.}$

De acuerdo al método de normalización propuesto por Dong *et al* [12], el valor de β se obtiene resolviendo la ecuación (2), donde la notación de u_{ij} son los momentos centrales:

$$u_{03}^{(1)}\beta^3 + 3u_{12}^{(1)}\beta^2 + 3u_{21}^{(1)}\beta + u_{30}^{(1)} = 0 \quad (2)$$

En este trabajo se considerara la solución analítica dada en [14]. Se utilizó este método porque permite encontrar la solución cuando ésta es única o hallar la del medio en el caso de haber tres soluciones reales, como requiere el método general propuesto en [12]. Las soluciones numéricas pueden a veces divergir o llevar a una solución que no es la buscada.

La nueva imagen obtenida se denomina $f_2(x,y)$.

c. Realizar una transformación de deformación en el eje ***y*** a la imagen $f_2(x,y)$. La matriz A_y que permite realizar la deformación vertical actúa como se muestra en la Figura 3.

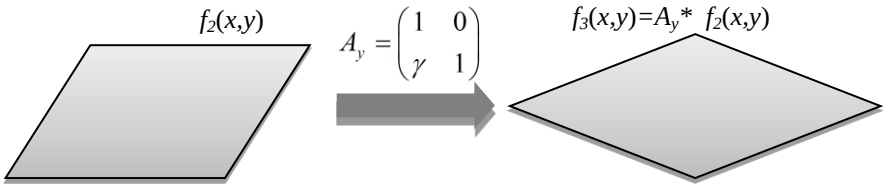

Figura 3 - Deformación vertical a $f_2(x,y)$.

De acuerdo a Dong *et al*, el valor de γ, se obtiene de la siguiente manera:

$$\gamma = -\frac{u_{11}^{(2)}}{u_{20}^{(2)}} \quad (3)$$

d. Realizar una transformación de escalamiento a la imagen $f_3(x,y)$ para obtener la imagen $f_4(x,y)$. Este paso permite modificar el tamaño de la imagen para que pueda entrar dentro del marco de trabajo como se ve en la Figura 4, aplicando la matriz A_s.

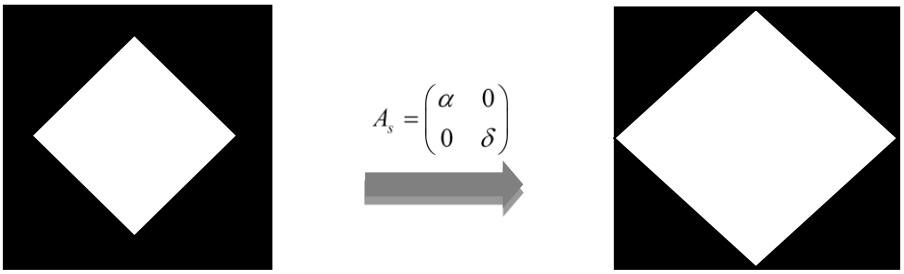

Figura 4 - Cambio de escala horizontal y vertical

Un ejemplo del procedimiento de normalización se muestra en la Figura 5.

2.2. Proceso de inserción

Para el proceso de inserción del watermark se efectúan los siguientes pasos:

i. Dividir la imagen normalizada $f_4(x,y)$ en bloques de 8 x 8 pixeles. En caso de que las dimensiones de la imagen no sean múltiplos de 8, no se consideran para la inserción los bloques que pertenecen al borde. Eso implica que el *watermark* se inserta solamente en los bloques 8x8 completos del interior de la imagen.
ii. Determinar bloques de textura fuerte (TF), es decir, seleccionar a aquellos que contengan bastante variación de colores dentro del bloque, para ello se siguen los siguientes sub-pasos.

   a. A cada bloque se aplica la Transformada Discreta de Coseno DCT (Discrete Cosine Transform), aplicando la Ecuación (4), de acuerdo a [15]:

(4)

$$D = T\,M\,T'$$

donde M es el valor de la matriz 8 x 8 obtenida sustrayendo 128 a cada pixel de la imagen original, es decir, la mitad del valor máximo posible de un pixel; T es la matriz ortogonal 8x8 de la ecuación (5) y T' es la transpuesta de T.

$$T_{i,j} = \begin{cases} 1/\sqrt{N} & \text{si} \quad i = 0 \\ \sqrt{2/N}\cos[\dfrac{(2j+1)i\pi}{2N}] & \text{si} \quad i > 0 \end{cases} \qquad (5)$$

Para $i,j=1,\ldots,8$. Como T es ortogonal, su inversa es también T'. Esto permite calcular la transformada inversa simplemente: $M = T'DT$.

b. Cuantificar los coeficientes de DCT de cada bloque, utilizando la matriz de cuantificación $Q(u,v)$, [16].

c. Para determinar si el bloque es de textura fuerte, éste debe verificar las siguientes condiciones:

$$1.\ F_k(0,0) > T_1$$
$$2.\ \#número\{[F_k(u,v)/Q(u,v)] \neq 0\} > T_2$$

donde $F_k(u,v)$ es la DCT del bloque 8x8, $0 \leq u, v \leq 7$. La notación [□] es la función de redondeo hacia 0, por ejemplo el valor 1,17 se redondea a 1. La función $\#número\{condición\}$, obtiene el número de elementos que satisfacen la condición. Los valores posibles son de 1 a 64, debido a que el tamaño del bloque es 8x8, haciendo un total de 64 coeficientes. $Q(u,v)$, es el valor de la matriz de cuantificación, la misma se utiliza para la compresión JPEG con el factor de calidad (quality factor, QF) de 50 que es el valor estándar utilizado por omisión en compresión JPEG. T_1 y T_2 son dos valores umbral predeterminados. En [2] se proponen $T_1 = 230$ y $T_2=13$. En este trabajo se utilizó el valor $T_2=10$ para identificar más bloques de textura fuerte y poder insertar más información porque un carácter ASCII insertado utilizará 8 bloques. Si un bloque cumple con las condiciones 1 y 2 entonces se dice que tiene alta textura. En estos bloques es más conveniente insertar información nueva que podrá pasar desapercibida por el SVH. En la Figura 5 se muestran los bloques (pequeños rectángulos en rojo) identificados como de textura fuerte.

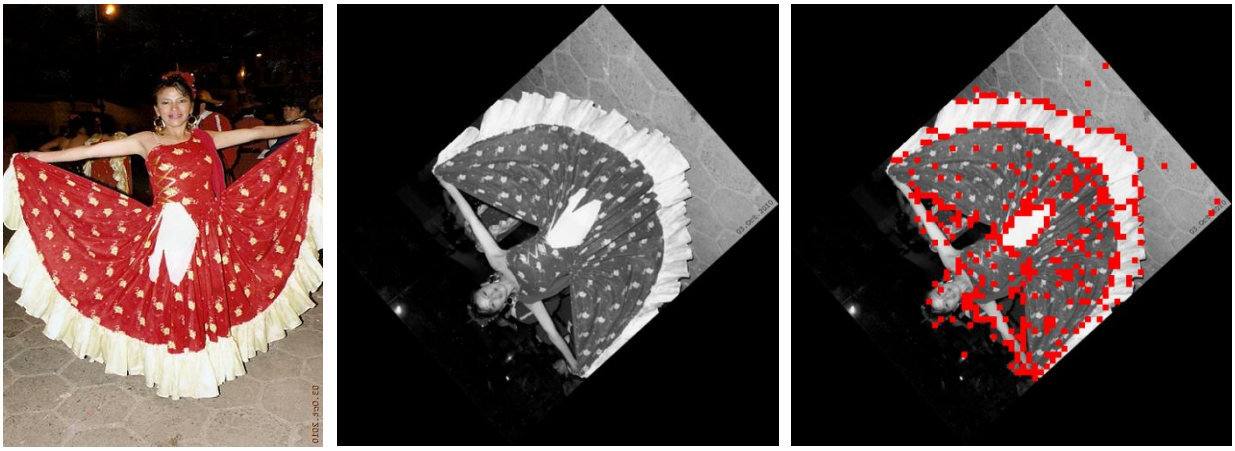

Figura 5 - Izquierda: Imagen original, Centro: normalizada, Derecha: bloques de textura fuerte.

iii. Realizar el proceso de cifrado del *watermark* ***W*** con Autómatas Celulares (AC). Para el cifrado del *watermark* el usuario debe insertar una clave ***K***, a partir de la cual se genera la cadena de bits que forma el estado en $t=0$ del autómata celular.

La evolución del autómata celular se realiza aplicando la regla de Wolfram [17] a cada celda:

$$\begin{cases} a_i^{(t+1)} = \left(a_{i-1}^{(t)} + a_i^{(t)} + a_i^{(t)} \square a_{i+1}^{(t+1)}\right) \bmod 2 \\ \qquad = a_{i-1}^{(t)} \text{ XOR } \left(a_i^{(t)} \text{ OR } a_{i+1}^{(t)}\right) \end{cases} \qquad (6)$$

Para cifrar el watermark se sigue el procedimiento estándar descrito en [18] cuyo resultado de la cadena de caracteres "SECRETO" es:

êÎdï?À?

iv. Generar un patrón bidimensional pseudoaleatorio llamado ***R***, cuyo tamaño es el mismo que la imagen original, posteriormente se genera el patrón ***MR*** que es el resultado de la normalización del patrón ***R*** como se ilustra en la Figura 6.

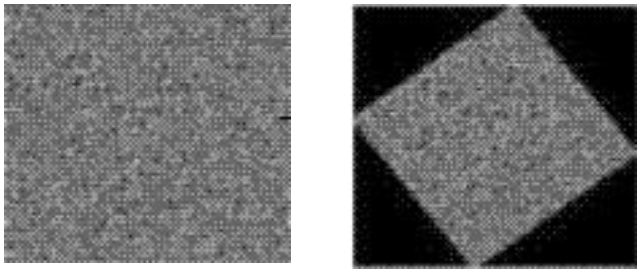

Figura 6 – Izquierda: patrón R, Derecha: patrón MR.

v. La cadena watermark ***W*** se inserta al patrón ***MR***, en la diagonal 4 [2] de cada bloque que se identificó como textura fuerte de la imagen original, por ejemplo:

La cadena de caracteres "Luz" con una clave "123" es:

$$\boldsymbol{W} = 01110110\ \ 11101100\ \ 10111111$$

El primer bit 0 es insertado en el primer bloque en la diagonal 4 que corresponde a la frecuencia de banda media , el siguiente bit 1 en el segundo bloque también en la diagonal 4, como se ilustra en la Figura 7.

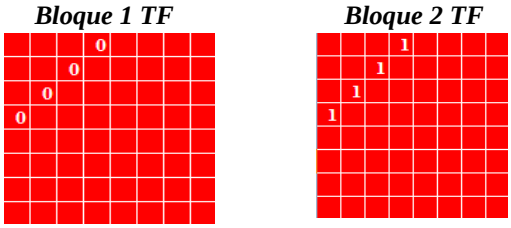

Figura 7 - Marcar Patrón MR.

En la práctica se utiliza un valor suficientemente negativo para representar el valor 0 y un valor suficientemente positivo para el valor 1. El patrón que contiene el *watermark* se denomina $\boldsymbol{MR_W}$.

Los bloques 8x8 se enumeran en zig-zag [16] de acuerdo a su codificación por entropía. Los 22 ***coeficientes de frecuencia de banda media,*** corresponden a los elementos de orden 7 al 28, en estas posiciones es donde inserta la cadena *watermark*. Los ***coeficientes de frecuencias altas*** (orden mayor a 28), generalmente son eliminadas con la compresión JPEG como si fueran *ruido* poco observable por el ojo humano. Los ***coeficientes de bajas frecuencias***, no se tocan porque el ojo humano es muy sensible a esas frecuencias y el *watermark* ya no sería imperceptible.

vi. Aplicar la Transformada Discreta del Coseno Inversa IDCT, al patrón $\boldsymbol{MR_W}$ que contiene la señal de *watermark* al que se denominará ***WP***.

vii. Aplicar el proceso inverso de normalización de imágenes al patrón ***WP.***

viii. El patrón de watermark ***WP*** se adiciona a la imagen original usando una inserción aditiva en el dominio espacial. La fórmula de la inserción está dada en la Ecuación (7) que es una modificación de la fórmula (13) de [2] para asegurar que no haya saturación del nivel de intensidad al añadir el watermark.

$$I_w = (1 - \alpha_2) * I_o + \alpha_2 * WP \qquad (7)$$

donde $\boldsymbol{\alpha_2}$ es un coeficiente llamado energía de inserción del watermark, que va a tomar valores entre 0 y 1.

[2] Si se denota una matriz *nxn* como *A(i,j)*, con *i,j=0…,n-1*, la diagonal 4 corresponde a los valores tales que *i+j=3*.

El proceso de inserción se resume en la Figura 8.

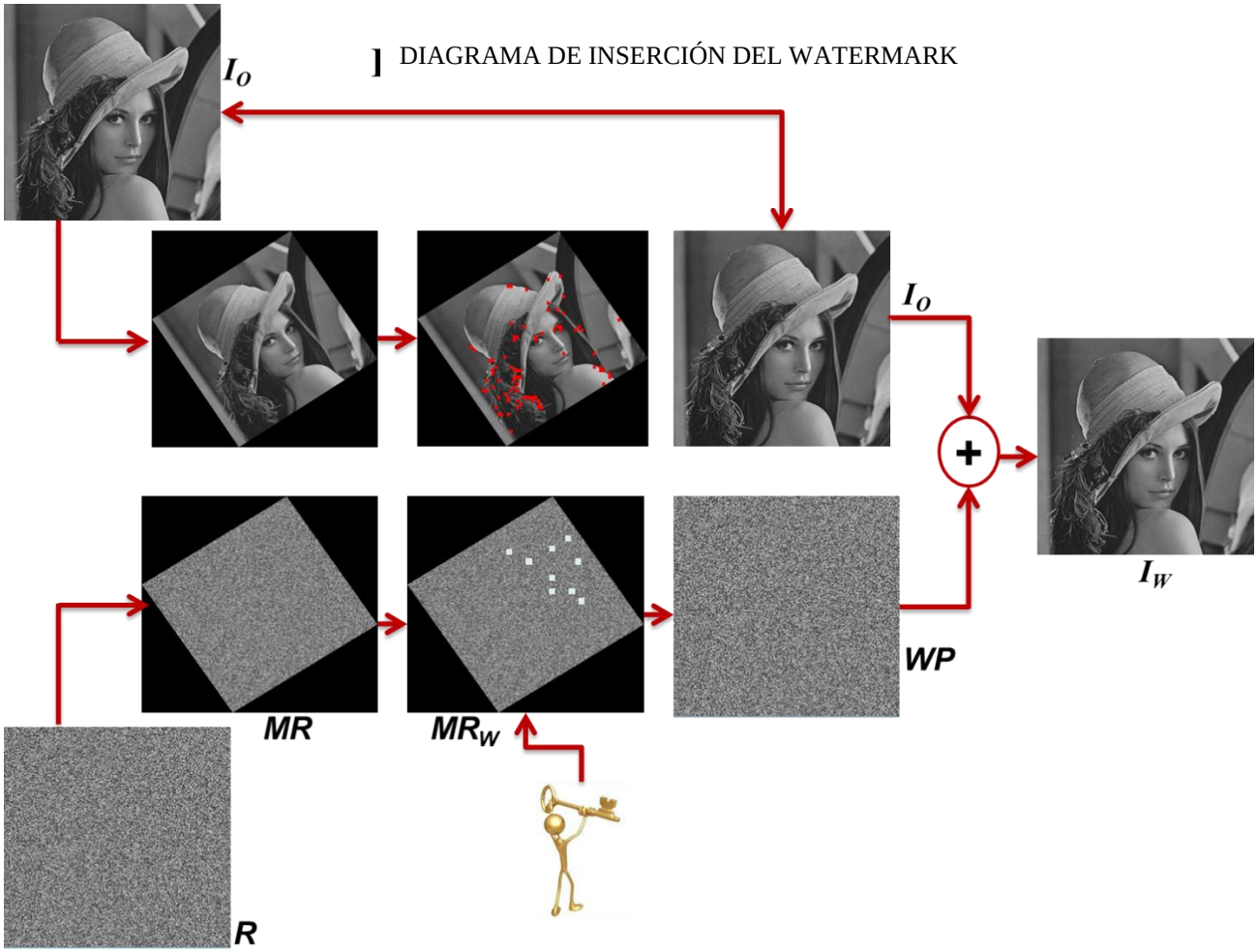

Figura 8 - Proceso de Inserción.

2.3. Proceso de extracción del watermark

El proceso de extracción del watermark se describe como sigue:

i. Se aplica el proceso de normalización a la imagen marcada.

ii. Se clasifica en bloques de 8x8 a la imagen marcada normalizada, para identificar los bloques de textura fuerte y bloques con textura débil. Desde los bloques con textura fuerte, se extraen los coeficientes marcados localizados en el rango de frecuencia media para obtener un promedio. Si el valor obtenido es negativo se lo reconoce como bit 0 de lo contrario será 1. Los bits extraídos se concatenan para formar la cadena *C*.

iii. La cadena extraída está cifrada. Para obtener la información de la imagen marcada es necesario conocer la clave del usuario con la que inserto el watermark, pues a partir de dicha clave se realiza el proceso de descifrado aplicando el autómata celular que se utilizó en la inserción, realizando una operación de XOR entre *K* y *C*, obteniendo como resultado la cadena *M* [17]. El proceso de extracción se resume en la Figura 9.

DIAGRAMA DE EXTRACCIÓN DEL WATERMARK

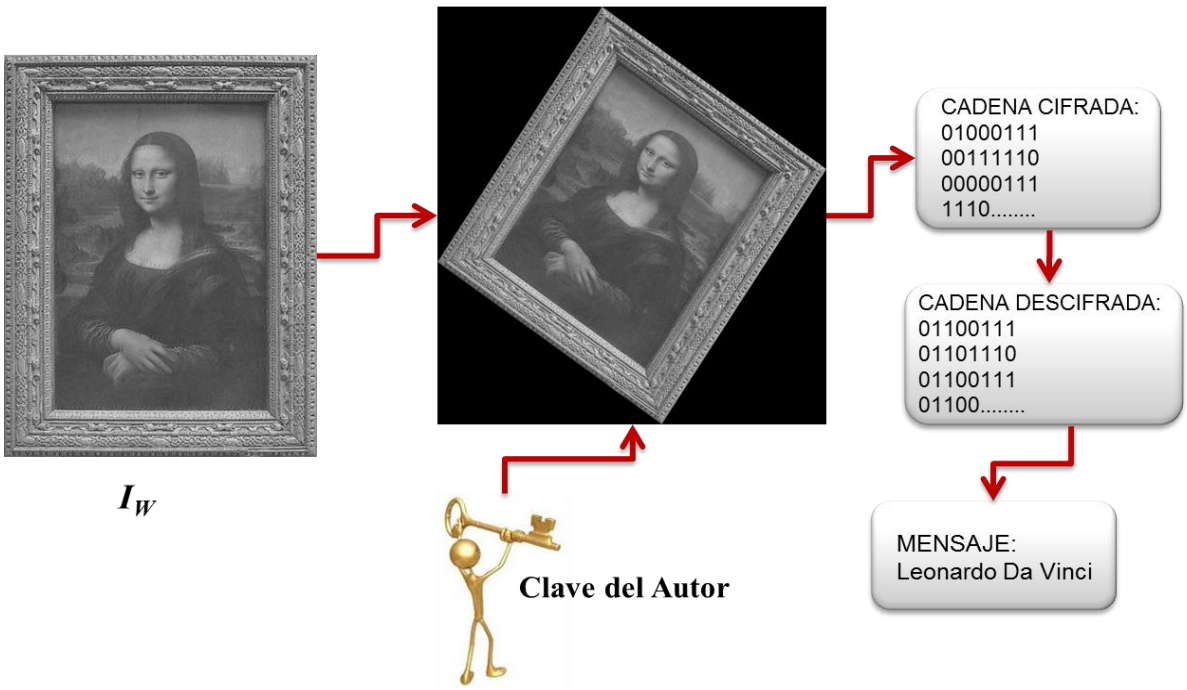

Figura 9 - Proceso de Extracción. I_w representa la imagen marcada que luego es normalizada, de la que se extrae el mensaje que contiene.

3. PRUEBAS Y RESUMEN DE RESULTADOS

Los procesos de normalización, inserción y extracción del watermark fueron implementados en el lenguaje C#. El prototipo permitió realizar pruebas con imágenes de 24 bits, en formato de mapa de bits (BMP) y JPEG y sólo requiere seguir una secuencia de botones, lo que puede ser realizado directamente por el autor que desea proteger sus derechos de autor de la imagen. Dos parámetros importantes para las pruebas son:

- **Imperceptibilidad del watermark**

  Para medir la relación entre el factor de energía $\boldsymbol{\alpha_2}$ y la imperceptibilidad de la señal del watermark, se aplica la Relación Señal a Ruido Pico (Peak signal-to-noise ratio, PSNR), definida en [3, fórmula 15], entre la imagen original y la imagen marcada. Un PSNR elevado indica una mejor imperceptibilidad.

- **Ambigüedad del watermark extraído**

  En el esquema de extracción es muy importante que la señal de watermark extraída no sea ambigua, sobre todo si se trata de un mensaje constituido por texto en ASCII. Considerando esta situación, la ***Tasa de Bits Erróneos (BER)*** de la secuencia binaria extraída respecto al mensaje insertado fue calculada a partir de los bits reconocidos incorrectamente.

3.1. Resultados ante ataques no intencionales

Compresión JPEG a diferentes niveles de calidad (QF). Los resultados de la Tabla 1 y Figura 10 muestran que BER aumenta cuando disminuye el QF de la compresión JPEG, en particular se observa que a partir del valor 60 % de QF la tasa de error es nula.

TABLA 1 - RESULTADOS DE TASA DE BITS ERRÓNEOS (BER) CON RESPECTO A RESISTENCIA A COMPRESIÓN JPEG

Energía α_2	PSNR	QF (%)	BER (%)
45%	22,278	100	0
		80	0
		60	0
		40	1,25
		20	6,25

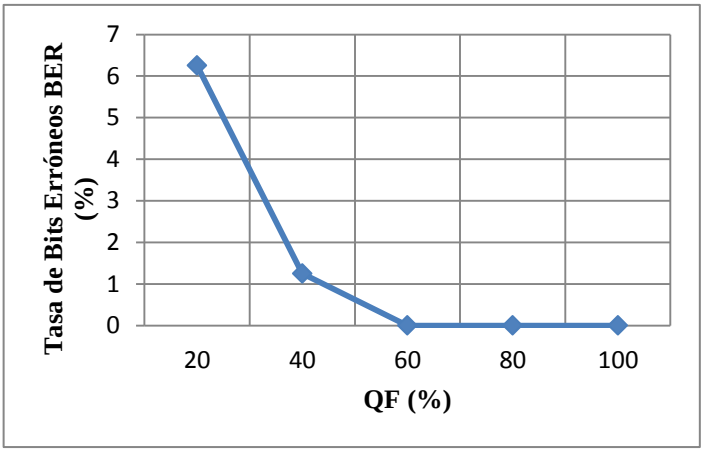

Figura 10 - Representación gráfica de la Tabla 1.

3.2. Resultados experimentales en relación a la energía

La Tabla 2 muestra los resultados con respecto a la variación de energía. Se observa en la Figura 11 que a mayor energía α_2 el PSNR va disminuyendo, lo que significa que la imperceptibilidad del *watermark* disminuye. En la Figura 12, a partir de aproximadamente 40 % de energia la tasa de bits BER es nulo, sin embargo, para valores menores a 30 se tiene un BER superior a 20 %.

TABLA 2 - RESULTADOS DE TASA DE BITS ERRÓNEOS (BER) CON RELACIÓN A LA ENERGÍA A_2

Imagen Marcada	Energía α_2 (%)	PSNR	BER (%)
En formato JPEG con QF 100%	10	35,2656	40
En formato BMP	10	33,8406	33,75
	20	27,5986	17,50
	30	24,0319	6,25
	35	22,6814	2,50
	36	22,4358	2,50
	37	22,1974	0
	38	21,9599	0
	40	21,4877	0
	50	19,5017	0
	60%	17,9346	0%

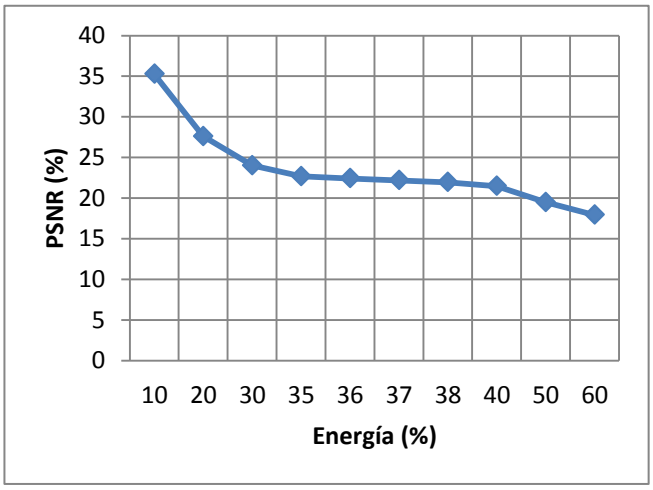

Figura 11 - Representación Gráfica de PSNR y Energía.

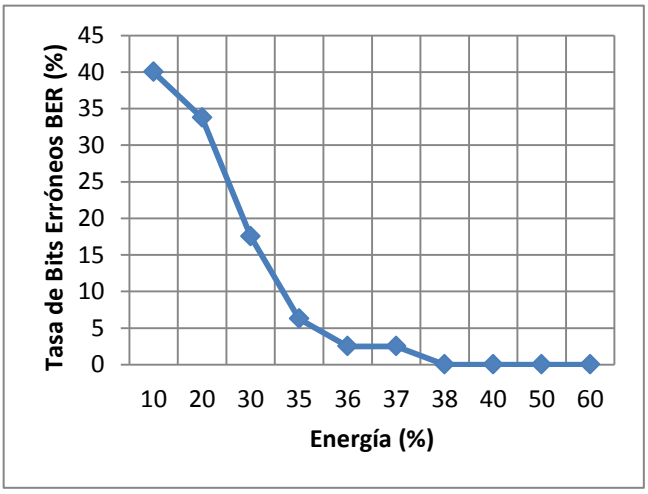

Figura 12 - Representación Gráfica de BER y Energía.

En la Figura 13 se muestra imágenes que tienen adicionado un watermark con diferente energía.

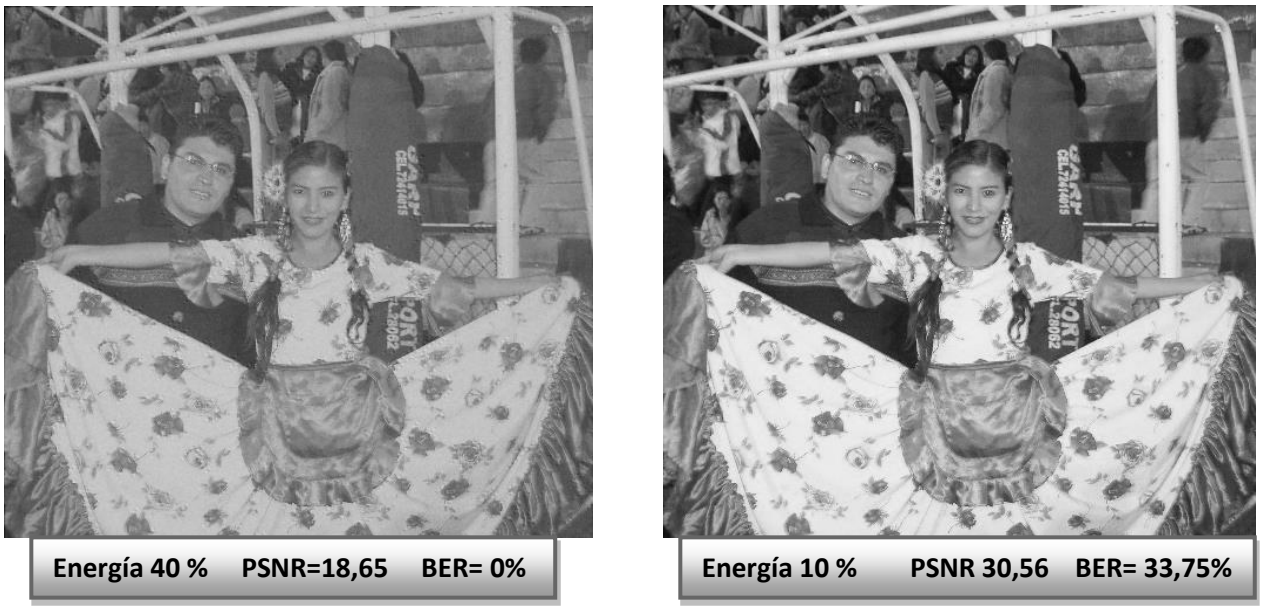

Figura 13 - Visibilidad del watermark para dos valores de la energía.

3.3. Resultados experimentales frente a ataques (ruido)

Para la verificación de resistencia del watermark frente a ataques como son la adición y filtros de ruido, se trabajó con una imagen marcada en un formato de mapa de bits BMP, con una energía de 40%.

En la Figura 14, el mensaje que contiene la imagen I_W son las iniciales del pintor Leonardo Da Vinci, i.e. "**ldv**", que corresponde a la cadena de bits cifrada.

01011000 10110110 00111101

Imagen marcada I_w con una energía de 40 %

Figura 14 – Izquierda: Imagen sin ruido, Derecha: imagen con ruido gaussiano.

De la imagen I_W con ruido gaussiano se extrae el mensaje con una tasa de error = 0%, de la imagen I_W con filtro gaussiano se extrae el mensaje con una tasa de error = 0% y de la imagen I_W con ruido salt-and-pepper, se extrae el mensaje con una tasa de error = 0%, sin embargo, reconoce un bit por demás, como se muestra a continuación:

01011000 10110110 001111101**1**

De la imagen I_W con filtro salt-and-pepper, se extrae el mensaje con una tasa de error 1/24=4.16%.

01011000 10110110 0**1**111101

La Figura 15 muestra la imagen marcada, denomina I_w , con los ataques mencionados anteriormente, de los cuales se extrae el mensaje insertado de manera no ambigua, legible para identificar la autoría de dicha imagen.

Figura 15 – Izquierda: I_w con filtro gaussiano, Centro: con ruido salt-and-pepper, Derecha: con filtro *salt-and-pepper*.

4. CONCLUSIONES

A partir de los resultados experimentales realizados con la presente aplicación de la técnica digital *watermarking* para la autenticación de imágenes digitales, se ha logrado insertar el *watermark* en una imagen de manera casi imperceptible añadiendo un nivel de seguridad mediante cifrado empleando el método de Autómatas Celulares. El *watermark* se recupera con una tasa de error próxima a cero cuando la imagen marcada no sufre ningún ataque o deformación. El *watermark* es resistente a la compresión JPEG, sin embargo la cadena extraída presenta una tasa de error que depende del factor de calidad de la compresión. El *watermark* con una energía del 40% es resistente al ruido Gaussiano, al filtro Gaussiano y al ruido *salt-and-pepper*, sin embargo, presenta una tasa de error pequeña al filtro *salt-and-pepper*. Está abierta la posibilidad de seguir trabajando en las técnicas de *watermarking* con el objetivo de reducir la energía de inserción y las tasas de error para hacer frente a todos los ataques, tanto intencionales como no intencionales, e incluso a los diferentes filtros, así también la investigación de otros métodos de *watermarking* que sean resistentes a la compresión de la Transformada Discreta de Wavelets (DWT). Finalmente, en un próximo trabajo se busca añadir métodos de detección y corrección de error para reducir la tasa de bits erróneos.

5. AGRADECIMIENTOS

Los autores agradecen a Manuel Cedillo Hernández por aclarar varias dudas acerca de su artículo, a Sviatoslav Voloshynovskiy por proporcionar gentilmente las notas de su curso sobre seguridad multimedia y a Renato Villán por las conversaciones muy interesantes sobre el área. La primera autora agradece además a Juan Ramiro Villa, tutor de la tesis de grado quien está al origen de este trabajo.

6. REFERENCIAS

[1] I. J. Cox et al. *Digital watermarking and steganography*, San Francisco CA., Morgan Kaufmann, 2007.
[2] M. Cedillo et al. *A robust watermarking technique based on image normalization*, Rev. Fac. Ing. Univ. Antioquia no.52 Medellín Apr./June 2010.
[3] S. Voloshynovskiy et al. *Multimedia Security*, Curso de Master, Universidad de Ginebra, 2010.
[4] D. Stinson. *Cryptography: Theory and Practice*, Discrete Mathematics and Its Applications, CRC-Press, 1ra edición, 1995.
[5] T. Cover and J. Thomas. *Elements of Information Theory*, Wiley-Interscience, 2da edición, 2006.

[6] J.O. Ruanaidh and T. Pun. *Rotation, scale and translation invariant digital image watermarking*. Proc. ICIP' 97, Atlanta, vol. 1, 1997, pp. 536-539.
[7] S. Roy and E. Chang. *Watermarking Color Histograms*, Proc. ICIP 2004, vol. 1, 2004, pp. 2191-2194.
[8] Z. Fan and Y. Zhao. "Image Watermarking Resisting to Geometrical Attacks Based on Histogram." Proc. Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, 2006, pp.79-82.
[9] S. Pereira and T. Pun. "Robust template matching for affine resistant image watermarks." *IEEE Trans. On Image Processing*, vol. 9, 2000, pp. 1123-1129.
[10] P. Bas et al. "Geometrically invariant watermarking using feature points." *IEEE Trans. on Image Processing*, vol. 11, no. 9, pp.1014-1028, Sept. 2002.
[11] M. K. Hu. *Visual Pattern Recognition by Moment Invariants*. IRE Trans. On Information Theory. vol. 8. 1962, pp. 179-187.
[12] P. Dong et al. "Digital watermarking robust to geometric distortions." *IEEE Trans. On Image Processing*, vol. 14. 2005, pp. 2140-2150.
[13] J. Huang and Y. Q. Shi. "Adaptive image watermarking scheme based on visual masking." *IEEE Electronics Letter*, vol. 34, 1998, pp. 748-750.
[14] J.V. Uspensky. *Theory of Equations*, McGraw Hill, New York,Vol. 57, 1948, pp.46-57.
[15] K. Cabeen and P. Gent. *Image compression and the discrete cosine transform*, Math 45, College of the Redwoods, Gent – 1998.
[16] International Telecomunication Union, *Information technology digital compression and coding of continuous-tone still images requirements and guidelines, Recommedation T.81, ITU, 1993.*
[17] S. Wolfram. *Cryptography with cellular automata*, in: Advances in Cryptology: Crypto'85. 344. Proceedings, LNCS 218, Springer, 1986, pp. 429–432.
[18] L. Hernández et al. *Aplicaciones de los Autómatas Celulares a la Generación de Bits*, Bol. Soc. Esp. Mat. Apl., no. 21, 2002, pp. 65-87.