



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
MÉXICO

FACULTAD DE INGENIERÍA

INCREASE IN THE INFORMATION PAYLOAD FOR
STEGANOGRAPHY IN DIGITAL IMAGES

ARTÍCULO ESPECIALIZADO PARA PUBLICAR
EN REVISTA INDIZADA

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

ISMAEL RUFINO GRAJEDA MARIN

Asesores:

DR. JOSÉ RAYMUNDO MARCIAL ROMERO
M. C. Héctor Alejandro Montes Venegas

TOLUCA, ESTADO DE MÉXICO, MARZO DE 2019

Índice

1.	Introducción	4
2.	Carta aceptación	6
3.	Artículo para publicación	8
4.	Discusión	15

1. Introducción

El presente artículo se deriva de una investigación previa realizada en la facultad de ingeniería [8], en la cual se propuso y desarrolló un algoritmo de esteganografía en imágenes digitales que permitiera mejorar en varios aspectos los métodos previamente existentes. Se desarrolló una adaptación al método denominado "diferencia de píxeles por tres vías". Con el objetivo de apoyar en la experimentación se realizó un estudio e implementación del mismo, derivado de los resultados experimentales se desprendieron 3 nuevos algoritmos.

Inicialmente se detectó una parte del algoritmo original que tentativamente podría ser optimizado de tal forma que se planteó la idea de modelar esa parte usando un método heurístico. Dada la complejidad del problema esta idea fue prontamente desechada pues el espacio de búsqueda estaba definido como 2^{28} por cada bloque. Pronto se detectó que el espacio de búsqueda podía ser reducido usando algunas funciones que permitían condicionar y delimitar el problema, logrando así obtener un espacio de búsqueda de 2^8 por cada bloque. Este espacio nos permitió lograr implementar un primer algoritmo exhaustivo, el cual brindo buenos resultados superando a los métodos ya existentes. Una segunda mejora se propuso logrando con ello crear un segundo algoritmo haciendo uso de conceptos básicos de cálculo diferencial.

Ambos algoritmos mencionados anteriormente presentaban los mismos resultados tanto en capacidad de almacenar información como en la calidad de la imagen resultante, únicamente presentaban diferencia obvia y esperada en términos de tiempo de ejecución. Tras un análisis del método, se detectó una anomalía en los resultados obtenidos del cual se presentaba un interesante patrón, mismo que se aprovecha para así insertar aún mas información en las imágenes, sin tener que sacrificar calidad en la imagen, obteniendo un tercer algoritmo que posteriormente en la fase de experimentación logra demostrar de forma contundente su alta eficiencia.

3. Artículo para publicación

Increase in the information payload for steganography in digital images

Ismael R. Grajeda-Marin

Facultad de Ingeniería,

Universidad Autónoma del Estado de México,
Cerro de Coatepec s/n, Toluca, Edo. de Mexico, Mexico
Email: tione_210@hotmail.com

Héctor A. Montes-Venegas

Facultad de Ingeniería,

Universidad Autónoma del Estado de México,
Cerro de Coatepec s/n, Toluca, Edo. de Mexico, Mexico
Email: hamontesv@uaemex.mx

J. Raymundo Marcial-Romero

Facultad de Ingeniería,

Universidad Autónoma del Estado de México,
Cerro de Coatepec s/n, Toluca, Edo. de Mexico, Mexico
Email: jrmarcial@uaemex.mx

J. A. Hernández-Servín

Facultad de Ingeniería,

Universidad Autónoma del Estado de México,
Cerro de Coatepec s/n, Toluca, Edo. de Mexico, Mexico
Email: xoseahernandez@uaemex.mx

I. RESUMEN

En la esteganografía sobre imágenes digitales, los métodos de diferencia de píxeles suelen utilizar la diferencia entre dos valores de píxeles consecutivos para determinar la cantidad de bits de datos que se pueden insertar en cada par de píxeles. La ventaja de estos métodos es la cantidad total de datos que puede almacenar en una imagen. Sin embargo, estos algoritmos con frecuencia desbordan los valores de píxeles, lo que da como resultado una imagen de salida incorrecta. Para evitar este problema, se agregan varios pasos adicionales para ajustar esos valores, o simplemente los píxeles se consideran inutilizables para incrustar datos, por lo tanto, se ignoran. En este artículo se presenta una adaptación del método de diferenciación de píxel por tres vías y encontramos un valor de píxel óptimo para cada bloque de píxeles computado de modo que su diferencia contenga los datos de entrada máximos y no existan píxeles de desbordados.

II. INTRODUCCIÓN

La esteganografía digital es el conjunto de técnicas diseñadas para ocultar datos digitales (la carga útil) dentro de un objeto digital. Una diferencia de las áreas relacionadas, como la criptografía o la marca de agua es que, las técnicas de esteganografía apuntan a mantener la existencia de un mensaje sin ser detectado y a aumentar continuamente la cantidad de datos de entrada que se incorporarán [1].

En la esteganografía de imágenes digitales, las intensidades de píxeles se utilizan para ocultar los datos. Un enfoque común es utilizar algún tipo de método de inserción de bit menos significativo (LSB) [2]. Los métodos por LSB reemplazan b bits menos significativos de los píxeles de la portadora con el mismo número de bits de datos de carga útil. Cuantos menos bits se reemplacen, menos alterada será la imagen del portador, pero también la carga útil será menor. Algunas técnicas de sustitución de LSB han implementado un ajuste óptimo de píxeles para la incorporación de datos para reducir

la interrupción de la imagen del operador [3]. Otros métodos esteganográficos incluyen una variedad de transformaciones, así como técnicas de enmascaramiento y filtrado. Una revisión detallada a los métodos actuales están disponibles en las literaturas. [1], [4].

Con el objetivo de aumentar la cantidad de datos que puede transportar una imagen, se ha propuesto un conjunto de técnicas que utilizan la diferencia entre dos píxeles vecinos para ocultar los datos de entrada. Esta diferencia puede ser calculada en cualquier dirección vecina. Wu y Tsai [5] propusieron un método de diferenciación de valores de píxeles (PVD) que produce una stego imagen con datos de carga útil considerable y una calidad de imagen sustancial. A partir de entonces se han producido diversos enfoques basados en PVD [6], [7], [8].

Idealmente, la carga útil debe recuperarse utilizando solo los valores de píxel resultantes, y todos los píxeles de la imagen original deben utilizarse para incrustar datos con el fin de lograr una mayor carga útil. Sin embargo, muchos métodos de PVD producen píxeles de desbordamiento o subdesbordamiento (es decir, fuera del intervalo de rango válido) y deciden ignorar o ajustar de alguna manera los valores de píxeles resultantes. Esto, sin embargo, puede llevar a una carga útil más baja o incluir estrategias adicionales para recuperar los datos incrustados [5] Eso puede revelar la existencia de un mensaje oculto.

En este documento, adaptamos el método de diferenciación de valor de píxel por tres vías y encontramos un valor de píxel óptimo para cada bloque de píxeles computado de manera que su diferencia contenga los datos de entrada máximos. Nuestro método reduce el tamaño del espacio de búsqueda y calcula un conjunto mucho más pequeño de soluciones factibles. Además, se discuten dos estrategias más para aumentar aún más el tamaño de la carga útil incorporada. El método está diseñado de tal manera que las intensidades de píxeles resultantes nunca esta fuera del intervalo válido y utiliza todos los bloques

de píxeles para transportar datos de carga útil. Una serie de resultados experimentales muestran la viabilidad del método.

Comenzamos en la sección III cubriendo los conceptos básicos del método de diferencia de valor de píxeles. Sección IV presenta una descripción detallada de nuestros dos enfoques de algoritmo de optimización. Sección V presenta varios resultados experimentales, y la sección VI presentamos nuestras conclusiones.

III. PIXEL-VALUE DIFFERENCING (PVD)

El método PVD [5] asume que la carga útil es un flujo continuo de bits de entrada que representan cualquier tipo de datos digitales. El PVD incrusta datos utilizando la diferencia de intensidad de dos píxeles contiguos. La idea es modificar estos píxeles agregando una conversión decimal de algunos bits de datos de entrada de tal manera que se mantenga su diferencia de valor para preservar la calidad de la imagen. Las regiones en la imagen con mayores diferencias en las intensidades de píxeles pueden llevar más piezas de carga útil que otras. Esto suele ocurrir en las áreas con bordes evidentes y menos frecuentemente en las regiones más suaves. El método proporciona una buena capacidad de incrustación pero es propenso a ser detectado usando métodos basados en estadísticas [9].

Chang *et al.* [7] propuso una versión modificada del PVD denominado *Tri-way Pixel-Value Differencing* (TWPVD). Mientras que el PVD inserta datos en un solo par de píxeles, el TWPVD usa diferencias horizontales, verticales y diagonales (de ahí su nombre) en 2×2 bloques de píxeles para ocultar los datos de entrada, logrando así una mayor carga útil que el PVD en la imagen del portador. Un problema que surge en los métodos basados en PVD es que frecuentemente producen valores de píxeles de desbordamiento / subdesbordamiento. Estos píxeles son ajustados o ignorados por el método, lo que reduce la cantidad de píxeles disponibles para llevar la carga útil de datos [10], [7].

III-A. Tri-way Pixel-Value Differencing

El método de Diferenciación de Valor de Píxeles en Tres Vías (TWPVD, por sus siglas en inglés) fue diseñado para involucrar a más píxeles en el proceso de integración de datos [7]. El TWPVD divide la imagen del operador en bloques no superpuestos de 2×2 píxeles consecutivos. Se calculan tres valores de diferencia en cada bloque a partir de los valores de dos píxeles vecinos en tres direcciones distintas. La primera diferencia se calcula entre el píxel en la esquina superior izquierda, es decir, el pivote, y el píxel a su derecha. La segunda diferencia es entre el pivote y el píxel en la esquina opuesta, y la tercera también está entre el pivote y el píxel debajo de él. Cada diferencia pertenece a uno de un conjunto predefinido de intervalos de rango que, a su vez, determina el número de bits que se insertarán en cada par de píxeles. Cada intervalo de rango R_k tiene una base l_k y un techo u_k . Valor listado en la forma de una tabla de rango. La tabla de rango se ha diseñado simplemente calculando el ancho de cada intervalo usando una potencia de dos, ya sea para proporcionar una gran

capacidad o para proporcionar una alta imperceptibilidad. [5], [6]. Otros enfoques han diseñado la tabla de rango basada en el *perfect square number* [11], o ha optado por reemplazar por completo la tabla de rango con una función bien diseñada basada en las funciones de piso y techo [8].

Independientemente de cómo se producen estos intervalos de rango, el algoritmo TWPVD sigue estos pasos:

1. Calcula las diferencias $d_i = p_i - p_1$ dentro del bloque de píxeles $i \in \{1, 2, 3, 4\}$, donde

p_1	p_2
p_3	p_4
2. Localiza para cada d_i el rango k tal que $l_k \leq |d_i| \leq u_k$
3. Calcular la cantidad de bits de datos de entrada t_i a ser insertado en la diferencia i de el bloque p_i como a continuación:

$$t_i = \begin{cases} 0 & \text{si } i = 1 \\ \lfloor \log_2(u_k - l_k + 1) \rfloor & \text{otro caso} \end{cases} \quad (1)$$

4. Calcula la representación decimal b_i de los t_i bits
5. Un nuevo d'_i es calculado para cada d_i

$$d'_i = l_{k_i} + b_i \quad (2)$$

6. Más tarde, el TWPVD utiliza cada uno d'_i para calcular los valores de los píxeles resultantes p'_i usando un conjunto de reglas bien diseñadas [7]. Hemos adaptado el TWPVD al reemplazar estas reglas con una estrategia de *optimización* para determinar los mejores valores de píxeles que contienen la carga útil máxima.

Una mirada más cercana a este algoritmo, revela que también produce valores de píxel de desbordamiento / subdesbordamiento que simplemente se omiten como portadores de carga útil de datos. Pero aún, los autores de TWPVD [7] Parece que no se discute cómo el algoritmo de extracción sabe qué píxeles se están ignorando [8]. Esto es fundamental para garantizar la integridad del mensaje secreto.

IV. UN ENFOQUE DE OPTIMIZACIÓN PARA MODIFICAR EL TWPVD

Cualquier método PVD se puede ver como un problema de optimización de la siguiente manera: Dado d'_i y p_i , busque una solución p'_i sujeto a las siguientes condiciones:

1. El desbordamiento / subdesbordamiento se logra evitar con $0 \leq p'_i \leq 255$
2. La recuperación de los datos de la carga útil está sujeta a $d'_i = |p'_i - p_1|$, donde p'_i y p_1 ahora son variables para buscar cómo un problema de optimización que definiría la stego-imagen.
3. La distorsión de la imagen resultante debe estar sujeta a minimizar la función objetivo.

$$f(p_i, p'_i) = \sum_{i=1}^4 (p_i - p'_i)^2 \quad (3)$$

Sabemos que $p'_i = |d'_i|$ es una solución, por ejemplo cuando $p_1 = 0$, cumple con las condiciones 1 y 2, pero no cumple la condición 3 porque causa una distorsión importante

	PSNR		Time	
	OTPVD	FBTPVD	OTPVD	FBTPVD
Barbara	36.50	36.50	471.60	9.56
Airplane	38.90	38.90	458.35	10.02
Boat	38.19	38.19	394.81	8.35
Goldhill	38.73	38.73	323.85	17.72
Lena	39.34	39.34	366.10	11.80
Average	38.33	38.33	402.94	11.49

TABLE I

COMPARACIÓN ENTRE OPTIMAL-TWPVD Y UN SIMPLE ALGORITMO DE FUERZA BRUTA AGREGADA AL TWPVD. LA VENTAJA DE RENDIMIENTO DE TIEMPO ES CLARA.

en la imagen de stego resultante. No obstante, la solución muestra que existe al menos una solución para cualquier entrada dada.

Dado que hay 4 píxeles por bloque en el rango $[0.,255]$, podemos estimar fácilmente que el tamaño del espacio de búsqueda sea 2^{32} combinaciones posibles de valores de píxeles multiplicado por cada bloque de píxeles en la imagen a procesar. Estas soluciones tardan demasiado en ser exploradas de manera eficiente, como se muestra en la Tabla I.

Comparación entre Optimal-TWPVD y un simple algoritmo de Fuerza Bruta agregada al TWPVD. La ventaja de rendimiento de tiempo es clara.

Una alternativa es reducir el tamaño del espacio de búsqueda para que pueda ser explorado en tiempos útiles. Usando la ecuación de la condición 2 se deduce que

$$p'_i = \pm d'_i + p'_1 \quad (4)$$

Evidentemente, esto significa que podemos calcular p'_i usando las dos variables siguientes:

1. $\pm d'_i$ toma las diferentes combinaciones de signos para d'_i . Estas combinaciones son 8, porque d'_1 siempre es 0 y d'_2, d'_3, d'_4 solo puede tomar 2 diferentes valores: uno positivo y otro negativo de igual magnitud.
2. p'_1 debe estar sujeto a $0 \leq p'_1 \leq 255$. Esto significa que p'_1 solo puede tomar 256 valores distintos.

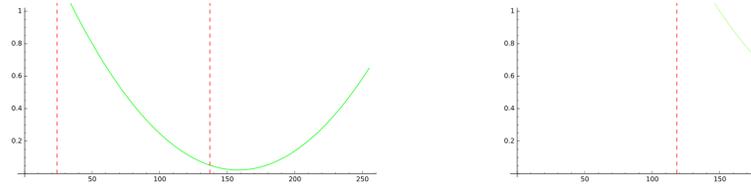
Esto reduce aún más el tamaño del espacio de búsqueda a 2^{11} . Un espacio de búsqueda de este tamaño se puede explorar fácilmente en su totalidad. Es decir, todos los valores posibles para p'_1 deben combinarse con todos los valores posibles para $\pm d'_i$.

IV-A. Una estrategia de optimización adicional

Ahora describimos una estrategia de optimización adicional para aumentar aún más la carga útil insertada por el método de la Sección IV. Dicha estrategia se basa en la primera derivada de la función objetivo con respecto a p'_1 y descarta las soluciones de desbordamiento.

Usando las ecuaciones 3 y 4, se puede producir una función cuadrática en términos de p'_1 , a saber:

$$f(p'_1) = \sum_{i=1}^4 (\pm d_i + p'_1 - p_i)^2 \quad (5)$$



a) Punto óptimo *fuera* el intervalo válido b) Punto óptimo *dentro* del

Fig. 1. Dos gráficos de función objetivo diferentes

Se pueden trazar ocho curvas cuadráticas diferentes obtenidas de las ocho diferentes combinaciones de signos en i . Cuando se calcula la primera derivada de estas funciones, se puede encontrar un punto para cada curva para el cual f es mínimo:

$$p'_1 = \frac{1}{4} \sum_{i=1}^4 p_i - \frac{1}{4} \sum_{i=1}^4 \pm d_i \quad (6)$$

Los valores candidatos de 8 para p'_1 pueden convertirse en 16 porque la ecuación (6) puede producir números reales que deben convertirse en números enteros utilizando tanto las funciones *ceil* como *floor*.

En algunos casos, el punto óptimo puede estar fuera del rango válido o incluso puede ocasionar que algunos de los otros 3 píxeles estén desbordados. Es necesario, entonces, mover ese punto dentro del rango apropiado, ya que ese valor es potencialmente una solución.

La figura 1 muestra 2 curvas trazadas usando la función objetivo. Estas curvas están limitadas entre un par de líneas de puntos que representan los límites superior e inferior válidos para p'_1 . También muestra que los puntos de valor mínimo no siempre están dentro del intervalo válido y es necesario mover ese punto a un área válida.

La ecuación (4) puede generar intervalos válidos para cada curva como $\max(\pm d_i) \leq p'_1 \leq \min(\pm d_i + 255)$. A partir de esta ecuación, podemos definir la función de ajuste:

$$A(p, M, m) = \begin{cases} 0 & \text{si } M \leq p \leq m \\ M - p & \text{si } p < M \\ -(p - m) & \text{si } p > m \end{cases} \quad (7)$$

Por lo tanto, el punto óptimo en el rango válido se definiría como:

$$p'_1 = p'_1 + A(p'_1, \max(\pm d_i), \min(\pm d_i) + 255) \quad (8)$$

Como se mencionó anteriormente, este punto debe ajustarse usando las funciones *ceil* o *floor*. Ambas funciones producen un valor idéntico o extremadamente cercano. Debido a que existen 8 cada una con 2 soluciones, terminamos con un nuevo espacio de búsqueda de solo 16 soluciones potenciales.

El algoritmo sigue estos pasos:

1. Ir a través de los pasos 1-5 del algoritmo de la sección III-A
2. Calcula $s_i = \pm d_i + p'_1$ usando las funciones *ceil* o *floor*. Las soluciones con desbordamiento se descartan

3. La solución óptima está dada por $p'_i = \min(f(p_i, s_i))$
4. Reemplace el bloque de píxeles original 2×2 con la solución óptima encontrada
5. Repetir para cada bloque de píxeles de 2×2 de la imagen del operador

Para recuperar el mensaje secreto, el proceso inverso se aplica de la siguiente manera:

1. Divida la imagen del operador en bloques no superpuestos de 2×2 píxeles consecutivos
2. calcula las diferencias $d_i = p_i - p_1$ dentro del bloque de píxeles con $i \in \{1, 2, 3, 4\}$
3. Para cada d_i localiza en la tabla de rangos $r_i = k$ tal que $l_k \leq |d_i| \leq u_k$
4. Calcula el número de bits a insertar en cada diferencia $t_i = \begin{cases} 0 & \text{si } i = 1 \\ \lfloor \log_2(u_{r_i} - l_{r_i} + 1) \rfloor & \text{otro caso} \end{cases}$
5. Toda la carga útil de datos se recupera concatenando la representación binaria de $b_i = d_i - l_{r_i}$

IV-B. Incrementando el bpp 0.25 puntos

El método puede insertar un bit adicional para aumentar aún más el mensaje secreto insertado en cada bloque 2×2 con un deterioro mínimo de la imagen del operador.

Las funciones *floor* y *ceil* producen dos números enteros consecutivos que resultan en la Función Objetivo muy cercanos o incluso idénticos. Este tipo de curvas aparecen constantemente y se utilizan como indicación para insertar un bit adicional del mensaje secreto. Este bit adicional se llama β . Si $\beta = 0$, p'_1 debe ser par, si $\beta = 1$, p'_1 debe ser impar.

Para encontrar el óptimo, decimos que $2c = p'_1 - \beta$ y modificamos la ecuación 4 de la siguiente manera:

$$p'_i = \pm d_i + 2c + \beta \quad (9)$$

$$f(c) = \sum_{i=1}^4 (\pm d_i + 2c + \beta - p_i)^2 \quad (10)$$

Por lo tanto, el intervalo válido para el problema de optimización viene dado por:

$$c = \frac{1}{8} \sum_{i=1}^4 p_i - \frac{1}{8} \sum_{i=1}^4 \pm d_i - \frac{1}{2} \beta \quad (11)$$

$$c = c + A(c, \max(-\frac{\beta}{2} - \frac{1}{2}(\pm d_i)), \min(-\frac{\beta}{2} - \frac{1}{2}(\pm d_i) + 255)) \quad (12)$$

El algoritmo también se modifica de la siguiente manera:

1. Repase los pasos 1-5 del algoritmo de la Sección III-A
2. Compute $s_i = \pm d_i + c$ usando las funciones *ceil* o *floor*. Las soluciones con se descartan
3. La solución óptima está dada por $p'_i = \min(f(p_i, s_i))$
4. Reemplace el bloque de píxeles original 2×2 con la solución óptima encontrada
5. Repetir para cada bloque de píxeles de la imagen del portador.

	bpp		PSNR		Time	
	OTPVD	POTPVD	OTPVD	POTPVD	OTPVD	POTPVD
Barbara	2.54	2.79	36.50	36.43	9.56	16.21
Airplane	2.37	2.62	38.90	38.76	10.02	17.99
Boat	2.41	2.66	38.19	38.09	8.35	20.24
Goldhill	2.38	2.63	38.73	38.64	17.72	16.53
Lena	2.35	2.60	39.34	39.17	11.80	15.65
Average	2.41	2.66	38.33	38.22	11.49	17.32

TABLE II
COMPARACIÓN ENTRE EL OPTIMAL-TWPVD Y EL OPTIMAL-TWPD CON INSERCIÓN DE BIT ADICIONAL

Para recuperar la carga útil del mensaje, se utilizan los mismos pasos de la Sección IV-A, y se agrega un bit adicional de 0 al mensaje si p_1 es par o 1 de lo contrario.

V. EXPERIMENTACIÓN

Se utilizó un conjunto de imágenes para probar el rendimiento de nuestros algoritmos y comparar nuestros resultados con los publicados anteriormente en la literatura. Todas las imágenes de portadora, que se muestran en la Figura 2, son imágenes en escala de grises de 8 bits de tamaño 512×512 . Muchas de estas imágenes se han convertido en un estándar *de facto* en experimentos de Procesamiento de imágenes y Visión artificial para probar nuevos desarrollos. También elegimos esas imágenes para comparar nuestros resultados con trabajos previos [12], [8]. Ambos autores compararon sus propios resultados con trabajos previamente publicados. Además, también comparamos el rendimiento de nuestro algoritmo con los resultados de TWPVD [7].

La relación pico de señal a ruido (PSNR) se utiliza para medir la diferencia entre la imagen original y la estego imagen. Cuanto más alto sea el PSNR, mejor será la calidad de la stego imagen. El número de bits por píxel (bpp) para cada imagen de prueba se calcula simplemente dividiendo el número de bits insertados por el número de píxeles en la imagen del portador.

La tabla II muestra una comparación entre los algoritmos Optimal-TWPVD y Extra Bit Insertion. Mientras que el primero muestra un mejor rendimiento que el trabajo anterior (como se muestra en Tablas V), el segundo aumenta aún más los resultados generales en términos de la cantidad de datos de carga útil (el *bpp*) insertado y la distorsión de la imagen medida con el PSNR en todas las imágenes probadas. Esto puede parecer esperado ya que las estrategias de TWPVD óptimo y de inserción de bit adicional utilizan cada bloque de 2×2 para llevar la carga útil de datos. Ningún bloque de píxeles se ignora y no se produjo ningún desbordamiento / subdesbordamiento de píxeles.

También comparamos nuestros resultados con los de TWPVD [7] en la Tabla III. Dado que nuestros algoritmos buscan los valores de píxel óptimos para cada bloque, los resultados son superiores en términos de *bpp* y PSNR. La noción general es que una menor cantidad de datos incrustados debería resultar en una menor distorsión de la imagen del



Fig. 2. Imágenes originales (primera fila). Stego-Imágenes resultantes utilizando el TPVD óptimo (segunda fila). Stego Imágenes resultantes con el uso de OTWPD y Extra Bit Insertion (tercera fila)

	bpp		PSNR		Time	
	TPVD	OTPVD	TPVD	OTPVD	TPVD	OTPVD
Barbara	2.54	2.54	36.38	36.50	1.55	9.56
Airplane	2.37	2.37	38.23	38.90	1.90	10.02
Boat	2.40	2.41	37.72	38.19	1.88	8.35
Goldhill	2.38	2.38	38.09	38.73	1.84	17.72
Lena	2.35	2.35	38.61	39.34	2.19	11.80
Average	2.41	2.41	37.81	38.33	1.87	11.49

TABLE III
COMPARACIÓN ENTRE EL TWPVD Y NUESTRO OPTIMAL-TWPVD

	bpp			
	OTPVD	POTPVD	H-Servin	F.Peng
Barbara	2.54	2.79	1.38	1.20
Airplane	2.37	2.62	1.30	1.20
Boat	2.41	2.66	1.80	1.20
Goldhill	2.38	2.63	1.66	1.20
Lena	2.35	2.60	1.60	1.20
airflid2	2.38	2.6	1.4	na
B2	2.4	2.7	1.6	na
baboon	2.45	2.68	1.65	1.2
Average	2.41	2.66	1.55	1.20

TABLE IV
COMPARACIÓN DE CARGA ENTRE NUESTRAS PROPUESTAS OPTIMAL-TWPVD EXTRA BIT OTPVD, Y HERNÁDEZ-SERVIN *et al.* [8] AND PENG *et al.* [12]

portador, lo que no se observa al comparar los valores de PSNR de nuestros experimentos.

Una comparación similar con los resultados recientes de Peng *et al.* [12] y Hernandez-Servin *et al.* [8] se muestra en la Tabla V. Esta tabla también muestra resultados favorables en términos de carga útil de datos transportada y calidad de la stego-imagen.

VI. CONCLUSIONES

Este trabajo diseña un algoritmo de optimización que modifica y mejora el método esteganográfico TWPVD [7]. Se compara favorablemente con el TWPVD y con los resultados recientes de Peng *et al.* [12] y Hernandez-Servin *et al.* [8].

	PSNR			
	OTPVD	POTPVD	H-Servin	F.Peng
Barbara	36.50	36.43	36.04	30.75
Airplane	38.90	38.76	36.09	33.45
Boat	38.19	38.09	34.56	26.66
Goldhill	38.73	38.64	37.03	30.70
Lena	39.34	39.17	37.55	26.89
Average	38.33	38.22	36.25	29.69

TABLE V
COMPARACIÓN DE PSNR ENTRE NUESTRAS PROPUESTAS OPTIMAL-TWPVD EXTRA BIT OTPVD, Y HERNÁDEZ-SERVIN *et al.* [8] AND PENG *et al.* [12]

Nuestros resultados muestran mejoras en varios aspectos importantes, a saber, (1) Número de bits por píxel insertado, (2) Mejor calidad de la estego-imagen medida con el PSNR, (3) No se producen píxeles de desbordamiento / subdesbordamiento, y (4) Sin bloques de píxeles omitidos o ignorados como portadores de datos.

El mérito principal de nuestros algoritmos es reducir el posible conjunto de valores de píxel posibles para cada bloque, de modo que la búsqueda de la mejor solución en términos de carga útil de datos y calidad de la estego-imagen se pueda realizar de manera eficiente.

Hay un par de direcciones en las que este trabajo puede proceder. El primer paso lógico es usar el método para ocultar la carga útil de datos en imágenes en color. El resultado obvio sería lograr una inserción de carga útil muy grande, pero los efectos en el color y la distorsión general de la imagen pueden requerir adaptar o cambiar por completo el algoritmo.

REFERENCES

- [1] Cheddad, A., Condell, J., Curran, K., Kevitt, P.M.: Digital image steganography: Survey and analysis of current methods. *Signal Processing* **90**(3) (2010) 727 – 752
- [2] Chan, C.K., Cheng, L.M.: Hiding data in images by simple lsb substitution. *Pattern recognition* **37**(3) (2004) 469–474
- [3] Wang, R.Z., Lin, C.F., Lin, J.C.: Image hiding by optimal lsb substitution and genetic algorithm. *Pattern recognition* **34**(3) (2001) 671–683
- [4] Mishra, M., Mishra, P., Adhikary, M.C.: Digital image data hiding techniques: A comparative study. *ANSVESA* **7**(2) (2012) 105–115
- [5] Wu, D.C., Tsai, W.H.: A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters* **24**(9) (2003) 1613–1626
- [6] Wu, H.C., Wu, N.I., Tsai, C.S., Hwang, M.S.: Image steganographic scheme based on pixel-value differencing and lsb replacement methods. *IEE Proceedings-Vision, Image and Signal Processing* **152**(5) (2005) 611–615
- [7] Chang, K.C., Chang, C.P., Huang, P.S., Tu, T.M.: A novel image steganographic method using tri-way pixel-value differencing. *Journal of multimedia* **3**(2) (2008) 37–44
- [8] Hernandez-Servin, J.A., Marcial-Romero, J.R., Muoz-Jimenez, V., Montes-Venegas, H.A.: A modification of the tpvd algorithm for data embedding. *Lecture Notes in Computer Science. Proc. of the Mexican Conference on Pattern Recognition* (2015)
- [9] Mahajan, M., Kaur, N.: Adaptive steganography: A survey of recent statistical aware steganography techniques. *International Journal of Computer Network and Information Security (IJCNIS)* **4**(10) (September 2012) 76–92
- [10] Mandal, J.K., Das, D.: Steganography using adaptive pixel value differencing (apvd) of gray images through exclusion of overflow/underflow. *CoRR* **abs/1205.6775** (2012)
- [11] Tseng, H.W., Leng, H.S.: A steganographic method based on pixel-value differencing and the perfect square number. *Journal of Applied Mathematics* **2013**(1) (2013) 1–8
- [12] Peng, F., Li, X., Yang, B.: Adaptive reversible data hiding scheme based on integer transform. *Signal Processing* **92**(1) (2012) 54 – 62

4. Discusión

En este trabajo se ha discutido una estrategia de optimización que modifica y mejora el método esteganográfico de diferencia de píxeles por tres vías TPVD[7]. Comparando su desempeño con el algoritmo TPVD, así como contra los resultados recientes de Peng et al. [12] y Hernandez-Servin et al. [8].

Los resultados experimentales permiten observar que los resultados de los algoritmos con las modificaciones propuestas muestran mejoras en varios aspectos importantes, a saber: (i) una mayor capacidad de carga útil, (ii) una mejor calidad de la stego imagen, (iii) el problema de desbordamiento a sido superado por completo, y (iv) no se ignoran bloques de píxeles como portadores de datos secretos.

En específico, los métodos de diferencia de píxeles proporcionaron una alta capacidad de inserción, así como una alta imperceptibilidad para las estego imágenes. tanto que el método TPVD optimizado mejora la seguridad y la calidad de la imagen resultante y evita el problema de desbordamiento. No obstante, el mayor mérito es la reducción del conjunto de posibles valores de píxeles para cada bloque, de manera que la búsqueda de la mejor solución en términos de mensaje secreto y calidad de imagen se pueda realizar de manera eficiente.

Después de revisar la riqueza de los diversos métodos basados en diferencia de píxeles reportadas en la literatura, hay un par de direcciones en las que la investigación de métodos PVD puede proceder. Primeramente, utilizar el método para ocultar datos en imágenes en color con el objetivo de lograr una imagen de más alta calidad. Esto ya se ha intentado con resultados mixtos, ya que los efectos en el color y la distorsión general de la imagen pueden requerir la adaptación o el cambio completo de los algoritmos de diferencia de píxeles para aumentar la capacidad y la imperceptibilidad.

Además, los métodos de diferencia de píxeles que utilizan varias direcciones de diferencia deben aumentar su imperceptibilidad y prueban su fuerza frente a los ataques de estegoanálisis. La mayoría de las técnicas publicadas muestran ventajas solo en términos de capacidad, una práctica común en el mismo, pero rara vez muestra resultados cuando está sujeto a métodos de estegoanálisis.

Una dirección de investigación más desafiante es el diseño de las tablas de rangos que determina el número de bits a incrustar. Ésta siempre está diseñada por el experimentador ya sea para lograr una gran capacidad o una alta imperceptibilidad. Este conjunto de los rangos tienen una importante influencia sobre el rendimiento de cualquier algoritmo de esteganografía basado en diferencias de píxeles. Se ha demostrado que dichos rangos pueden ser reemplazado por una sola función, aunque las habilidades de diseño del experimentador todavía son necesarias, por lo que, es una investigación abierta.