

COMPRESIÓN DE DATOS HOLOGRÁFICOS POR MEDIO DE ANÁLISIS DE FASE

Velez Zea, Alejandro^{1,2,3}; Villamizar Amado, Astrid Lorena^{1,2}; Tebaldi, Myrian^{1,3,4}; Torroba, Roberto^{1,3,4}

¹ Centro de Investigaciones Ópticas (CONICET La Plata- CIC-UNLP), P.O. Box 3, C.P 1897, La Plata, Argentina.

² Facultad de Ciencias Exactas, Universidad Nacional de La Plata, La Plata, Argentina.

³ Departamento de Ciencias Básicas, Facultad de Ingeniería, Universidad Nacional de La Plata, La Plata, Argentina.

⁴ UIDET OPTIMO, Departamento de Ciencias Básicas, Facultad de Ingeniería, Universidad Nacional de La Plata, La Plata, Argentina.

email: alejandrov@ciop.unlp.edu.ar

INTRODUCCIÓN

La holografía digital es una técnica con un amplio abanico de aplicaciones en las áreas de la física, la ingeniería y la medicina, entre otras. Sin embargo, a pesar de su utilidad demostrada, presenta múltiples retos que deben ser resueltos para permitir nuevos avances. Entre ellos, nos encontramos con el problema del manejo de grandes volúmenes de información producidos por los sistemas de registro holográfico. Esto implica un alto costo para el almacenamiento y transmisión de información relacionada con estos métodos, limitando así muchas técnicas, como las que requieren procesado de hologramas en tiempo real.

A diferencia de un registro fotográfico convencional, los registros holográficos contienen información tanto de amplitud como de fase. En particular, la parte esencial de los datos necesarios para la reconstrucción de un objeto o escena a partir de un holograma, están codificados en la fase extraída del mismo.

Esta fase tiene características que la hacen difícil de comprimir usando algoritmos tradicionales, como los empleados para el almacenamiento de imágenes (JPEG, LZW o DEFLATE). En particular, investigaciones recientes sobre el uso de técnicas ópticas para la compresión de datos holográficos [1–4], permiten concluir que las discontinuidades propias de las funciones de fase son las responsables del bajo rendimiento de estos algoritmos. Esto es debido a que los métodos digitales están optimizados para el tratamiento de funciones continuas.

En este trabajo, analizamos las discontinuidades de la fase y buscamos representaciones alternativas de la misma que permitan mejorar la compresión usando los métodos estándar. Para lograr este objetivo, generamos hologramas de fase computacionalmente, usando el algoritmo de Gerchberg-Saxton (GS) [5]. Posteriormente, proponemos una transformación basada en el desenvolvimiento parcial de la fase, lo cual permite una reducción de 58% del volumen de la información de fase tras la compresión, cuando se compara con la aplicación directa de los algoritmos a la fase sin procesar. Finalmente, verificamos que la nueva representación de la fase permite una reconstrucción fiel al objeto original.

Presentamos análisis teóricos y resultados numéricos que demuestran la validez de nuestra propuesta.

Generación de hologramas de solo fase con el algoritmo de Gerchberg-Saxton

Para poder demostrar la efectividad de nuestra propuesta, primero procedemos a generar hologramas de solo fase usando el algoritmo de GS. Los hologramas de solo fase, como su nombre lo indica, son aquellos en los cuales toda la información de un objeto esta codificada en la fase, con una amplitud uniforme o casi uniforme [6]. Estos hologramas tienen varias ventajas sobre los hologramas comunes, pues pueden ser reconstruidos ópticamente usando un modulador espacial de luz (MEL) de solo fase, y al no causar modulación en amplitud habilita un uso mas eficiente de la iluminación disponible durante la reconstrucción.

El algoritmo de GS permite encontrar la fase que conecta una medida de intensidad con la intensidad de la transformada de Fourier (TF) de la misma, lo cual es de gran utilidad pues registrar la intensidad de un campo óptico es mucho más sencillo que registrar su fase. Este mismo algoritmo nos posibilita generar hologramas de solo fase, simplemente introduciendo la intensidad del objeto cuyo holograma queremos generar y suponiendo que la intensidad de su TF es uniforme.

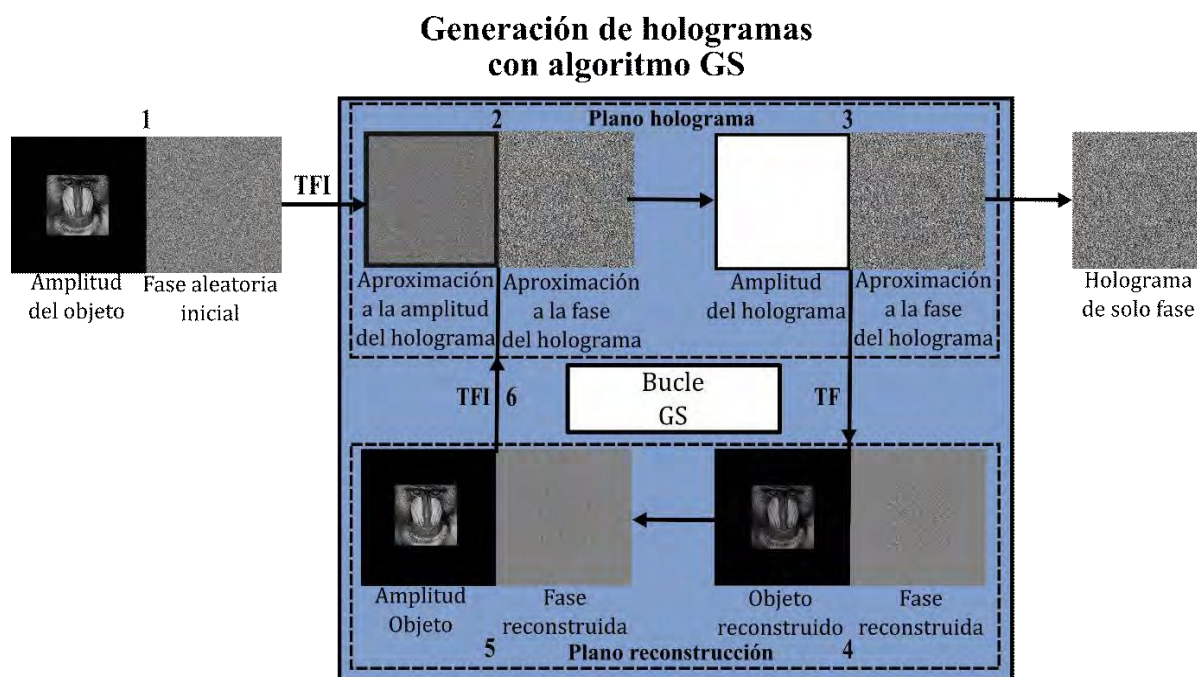


Figura 1: Diagrama del algoritmo de GS aplicado a la generación de hologramas de solo fase.

En la figura 1 se muestra el diagrama de algoritmo de GS aplicado a la generación de hologramas de solo fase. Este algoritmo se resume en los siguientes pasos.

1. Se multiplica la amplitud del objeto cuyo holograma se desea generar por una fase aleatoria inicial.
2. Se realiza la transformada de Fourier inversa (TFI) del producto obtenido en 1.
3. Se reemplaza la amplitud de la TFI obtenida en 2 por una amplitud uniforme.
4. Se realiza la TF del resultado de 3.
5. Se reemplaza la amplitud obtenida de 4 por la amplitud del objeto original.
6. Se realiza la TFI del resultado de 5.

Se repite el proceso hasta que la amplitud obtenida en 4 y la amplitud del objeto original sean lo suficientemente parecidos. Para determinar esto se usa una métrica de comparación, como el coeficiente de correlación. Una vez se cumpla esta condición, la fase obtenida en 3 en la última iteración corresponde al holograma de solo fase.

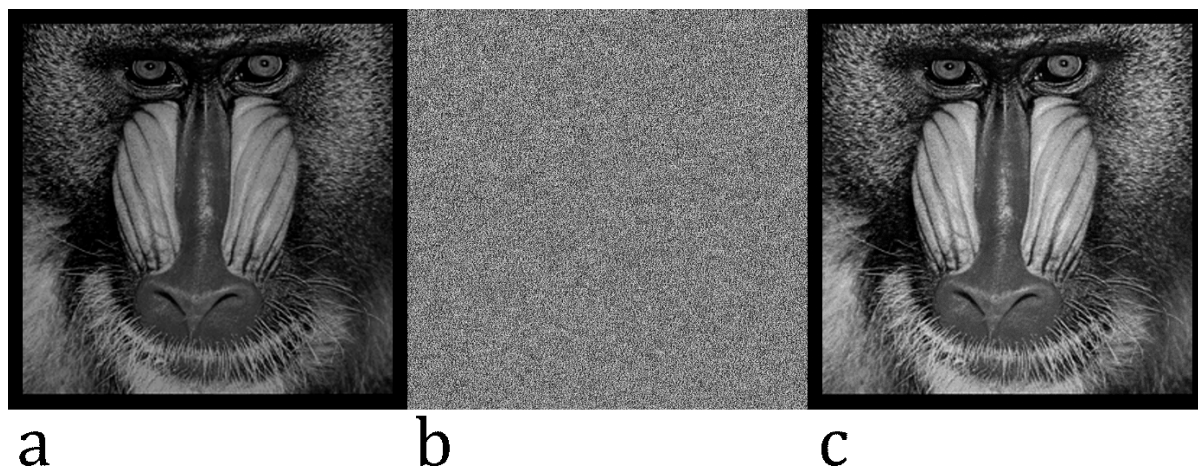


Figura 2: a) Objeto de entrada, b) holograma de fase, c) objeto reconstruido.

En la figura 2 se muestra el resultado de la generación de un holograma de solo fase de 1080x1080 píxeles de un objeto de 500x500 píxeles. La figura 2a corresponde al objeto original cuyo holograma queremos generar. El holograma resultante tras 100 iteraciones del algoritmo GS se muestra en la figura 2b. La figura 2c corresponde a la reconstrucción numérica del holograma. Como se puede apreciar, el holograma generado tiene una estructura altamente aleatoria. Esta aleatoriedad dificulta la compresión por medios digitales, debido a que la mayoría de los algoritmos busca redundancias en la información que permita representaciones de menor volumen.

Una forma de evaluar la aleatoriedad de una función consiste en calcular su entropía. Los hologramas generados en este trabajo son almacenados como una imagen con una profundidad de 8 bits, es decir, la fase puede tomar 256 valores distintos en el rango $0-2\pi$. De esta manera, la entropía está dada por

$$S = \sum_{i=1}^{256} p_i \log_2(p_i) \quad (1.1)$$

Donde p_i son las cuentas normalizadas del i valor del histograma de la imagen. Como referencia, recordemos que el máximo valor de la entropía para una imagen con 256 valores es 8, que corresponde a una distribución uniforme de valores en el histograma de la imagen. En la figura 3 se muestra el histograma correspondiente al holograma de la figura 2b. El histograma confirma el hecho de que este holograma es altamente aleatorio, y su entropía es de 7.998, cerca al máximo posible.

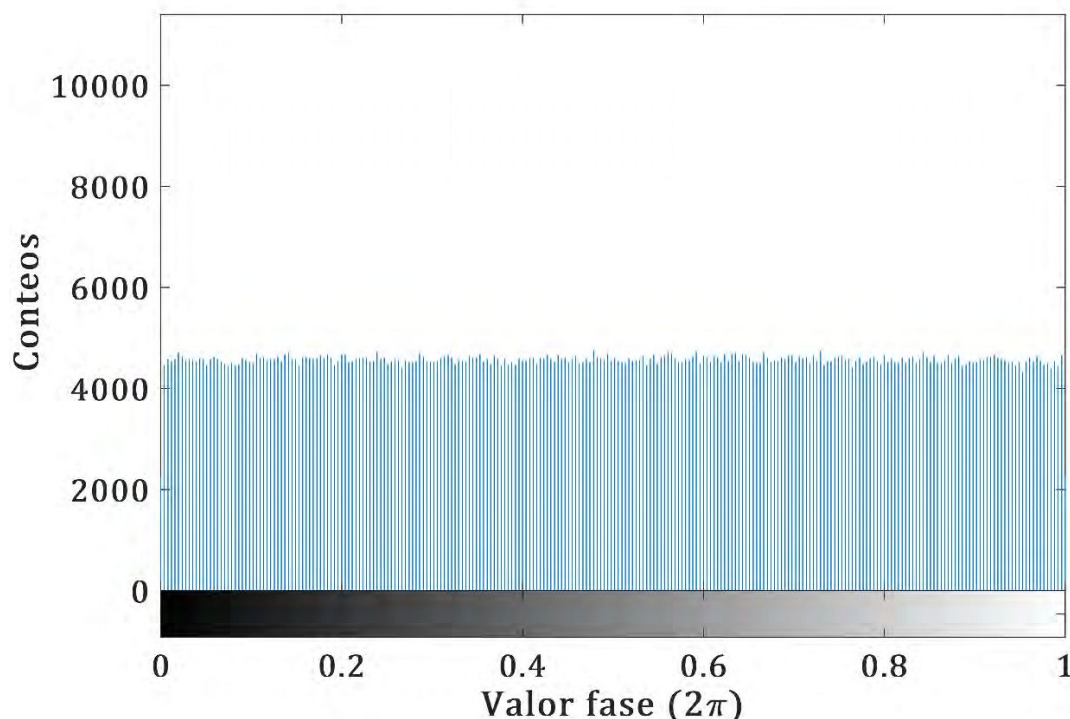


Figura 3: Histograma del holograma de solo fase de la figura 2b.

Representación alternativa de fase

La idea básica de nuestra propuesta es buscar una nueva representación del holograma, tal que este tenga menor entropía y pueda ser más susceptible a compresión sin pérdida. La hipótesis de la que partimos es que parte de la entropía de la información de fase es causada por el hecho de que es una función con discontinuidades cada 2π . Esto es debido a que la fase es el argumento de un exponencial imaginario. Teniendo esto en cuenta, eliminar parcial o totalmente las discontinuidades contribuirá a disminuir la entropía de la fase y lograr así mayor compresión [7].

La condición que buscamos para nuestra nueva representación de la fase es que esta permita reconstrucción del holograma sin pérdida. Esta condición se cumple si

$$e^{i\phi(v,w)} = e^{i\phi_a(v,w)} \quad (1.2)$$

Donde $\phi(v,w)$ es la fase original y $\phi_a(v,w)$ es la nueva representación.

Una de las formas de lograr una representación que cumple esta condición es usar técnicas de desenvolvimiento de fase. El desenvolvimiento de fase comprende a una serie de algoritmos que permiten convertir una función de fase en una función continua. Estos algoritmos son usados para obtener información acerca de objetos 3D en sistemas de holografía, radar y tomografía [8,9].

El desenvolvimiento completo de la fase, sin embargo, tiene la dificultad de que la función resultante puede tener un rango tal que no pueda ser almacenada de manera efectiva en una profundidad de bit determinada. Debido a este motivo, nuestra propuesta para lograr una representación alternativa de fase (RAF) permite seleccionar un rango de valores posibles para la fase, limitando las discontinuidades a valores que superen este rango. El rango de valores de la representación está dado por $2R\pi$, con R un número entero positivo. Los pasos del algoritmo para generar la RAF son los siguientes.

1. Tomar la primera columna de la matriz que contiene la fase.
2. Seleccionar el segundo píxel de la columna y calcular su diferencia con el primer píxel.
3. Si la diferencia es mayor a π , substrair 2π del resto de la columna. Si la diferencia es menor a $-\pi$, añadir 2π al resto de la columna.

4. Si el valor resultante del píxel actual es mayor que $R\pi$, substraer $2R\pi$ del resto de la columna. Si el valor es menor que $-R\pi$, añadir $2R\pi$ al resto de la columna.
 5. Repetir este proceso para cada píxel de la columna, comparando su valor con el píxel anterior.
 6. Repetir este proceso por cada columna del holograma.
- De esta manera, finalmente obtenemos una RAF $\phi_a(v, w)$ que satisface la condición de la Ec. 2.

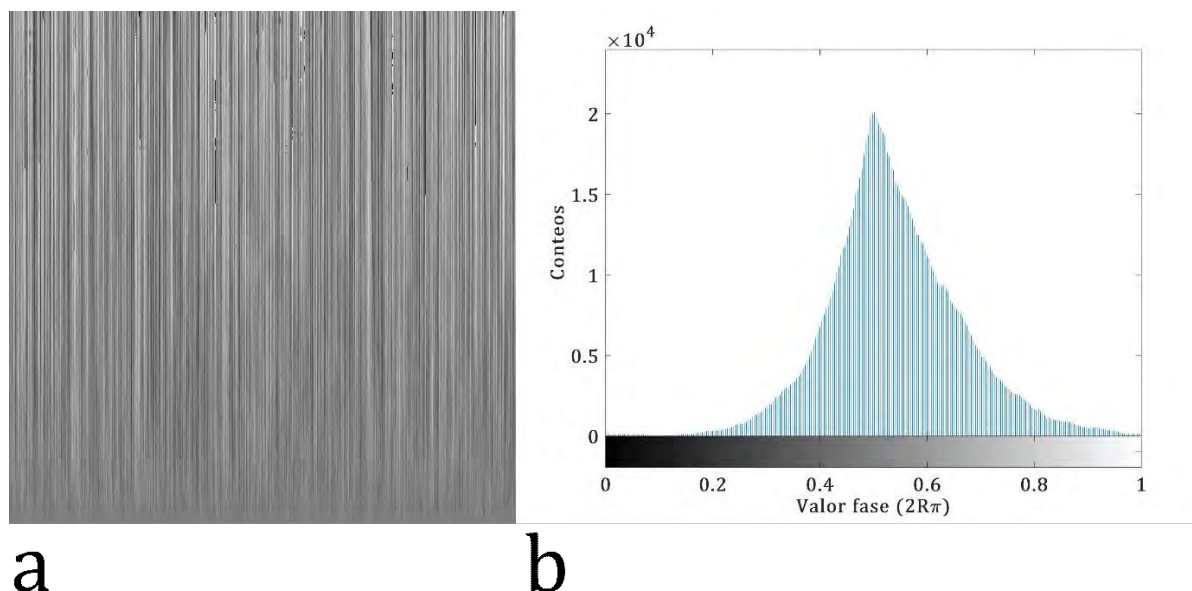


Figura 4: a) RAF del holograma de la figura 2b, b) histograma de a).

En la figura 4 se muestra una RAF obtenida a partir del holograma de la figura 2 con un rango de 54π , y el histograma correspondiente. Como se puede apreciar, la RAF tiene una estructura mucho menos aleatoria que el holograma original, con una entropía de 6.8594. El algoritmo usado para obtener la RAF permite controlar la entropía resultante variando el rango máximo, sin embargo, al tratar de almacenar funciones con mayores rangos en una imagen de 8 bits estas estarán sometidas a un mayor error de cuantización, resultando en una pérdida de calidad en la reconstrucción. Para mostrar este efecto, calculamos la entropía de las RAF obtenidas con distintos rangos máximos y el coeficiente de correlación entre los objetos reconstruidos a partir de estas RAF y el objeto reconstruido a partir del holograma original.

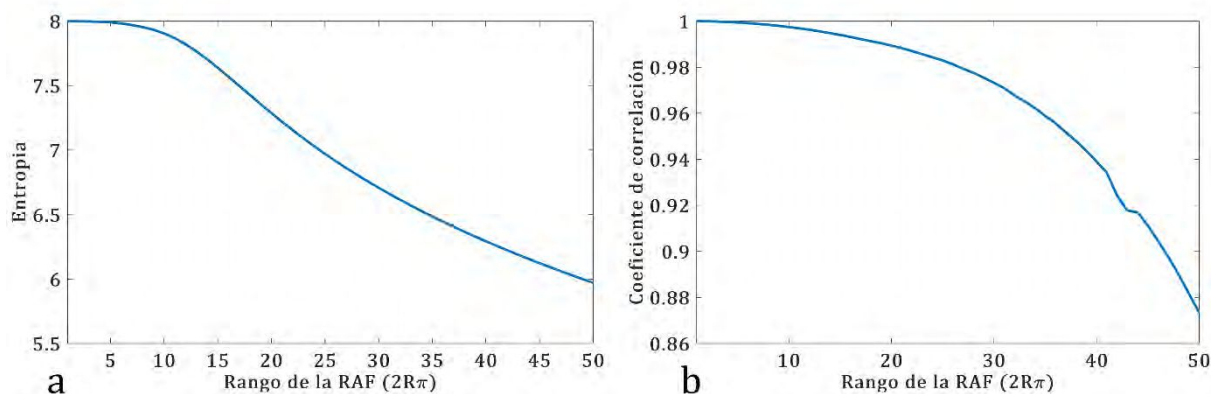
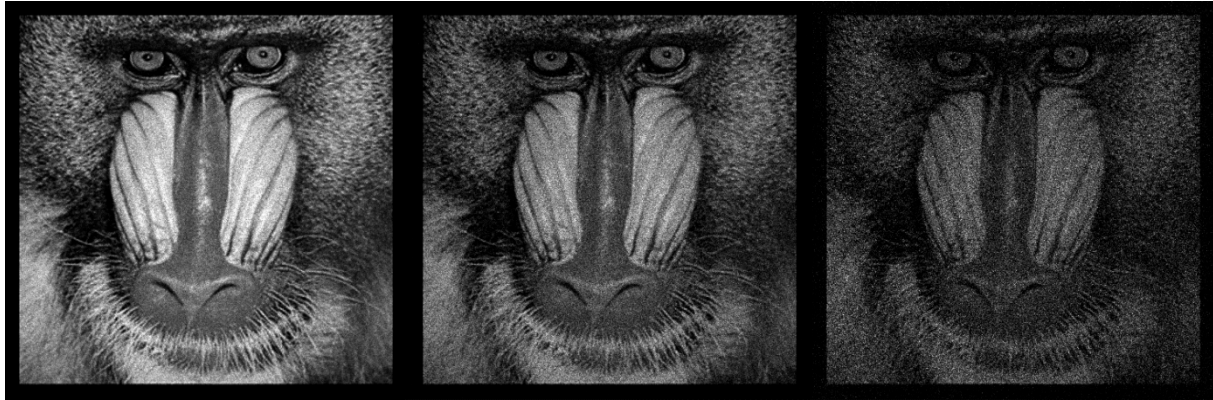


Figura 5: a) Entropía de las RAF obtenidas para distintos rangos, b) Coeficiente de correlación entre el objeto obtenido del holograma original y los obtenidos a partir de RAFs de distinto rango.

En la figura 5 se muestra el resultado de las pruebas mencionadas. Como se puede observar, tanto la entropía como el coeficiente de correlación disminuyen cuando se aumenta el rango de la RAF, lo que implica que hay una relación de compromiso entre lograr una mayor compresión y la calidad de la reconstrucción. Sin embargo, vale la pena mencionar, que la pérdida de calidad no es visualmente apreciable hasta que se llega a rangos elevados, con coeficientes de correlación por debajo de 0.9.



R=2

R=27

R=50

Figura 6: Objetos reconstruidos con RAFs de distintos rangos

Para respaldar esta afirmación, mostramos en la figura 6 el objeto reconstruido a partir de RAF con R=2,27 y 50, correspondiente a un rango de $4\pi, 54\pi, 100\pi$. Como se puede apreciar, el objeto reconstruido con R=27 no presenta una degradación importante, sin embargo, el obtenido con R=50 sufre de un nivel notable de ruido de speckle.

Compresión sin pérdida de RAF

Una vez demostrado el algoritmo usado para la generación de las RAF, ahora procederemos a probar la compresión que se obtiene aplicando algunos algoritmos comunes a RAFs con distintos rangos y comparando con la compresión que se logra aplicando esos mismos algoritmos al holograma original. Los algoritmos escogidos son el LZW (acrónimo de Lempel-Ziv-Welch, sus creadores), el DEFLATE y el PACKBITS, los cuales son usados comúnmente en la compresión de imágenes con el formato TIFF. El algoritmo LZW [10] es una mejora del algoritmo LZ77 [7] de compresión adaptativa, que busca grupos de caracteres o valores repetidos y los reemplaza con una palabra código que representa todo el grupo.

El algoritmo DEFLATE es una combinación del algoritmo LZ77 con la codificación Huffman [11], que permite usar palabras código de longitud variable para representar grupos de caracteres repetidos. Finalmente, PACKBITS es un algoritmo simple de codificación de longitud de grupo [12], que simplemente busca grupos de caracteres repetidos y los reemplaza por el carácter y el número de veces que se repite en el grupo.

Usando estos tres algoritmos, comprimimos el holograma original y las RAF, y calculamos el factor de compresión F definido como

$$F = \frac{V_i}{V_f} \quad (1.3)$$

Donde V_i es el volumen de datos de la fase original sin comprimir, y V_f el volumen obtenido tras la compresión.

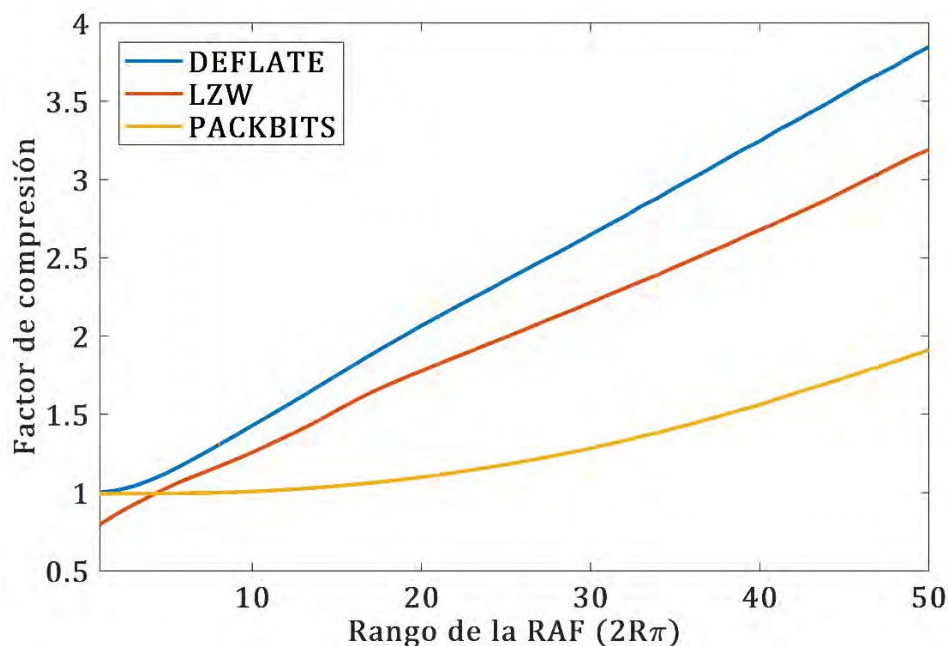


Figura 7: Factor de compresión de las RAF obtenidas para diferentes rangos tras aplicar los algoritmos DEFLATE, LZW y PACKBITS.

En la figura 7 se muestra el resultado de la prueba propuesta. Tal y como podíamos inferir de los resultados de la figura 5a, un mayor rango implica menor entropía, lo que se traduce directamente en un aumento del factor de compresión logrado con todos los algoritmos probados. En particular, el algoritmo DEFLATE permite obtener el mayor factor de compresión para todas las RAF probadas, lo que es de esperarse ya que incorpora tanto codificación Huffman como el algoritmo LZ77. El resultado de comprimir el holograma original se corresponde al resultado con rango $R=1$, y da lugar a un factor de compresión de 1 para DEFLATE, de 0.996 para PACKBITS y de 0.792 para LZW. Estos factores indican que el holograma original no se reduce en volumen al ser sometido a estos algoritmos debido a su altísima entropía, e incluso puede resultar con un volumen mayor al original, debido a la inclusión de los diccionarios de palabras clave que genera el algoritmo LZW.

CONCLUSIONES

En este trabajo hemos demostrado como es posible definir representaciones alternativas de información de fase con menor entropía que la presente en un holograma de fase generado con el algoritmo de GS. Mostramos como una RAF que disminuya las discontinuidades de la fase resulta en una representación más susceptible de ser comprimida con algoritmos standard. La relación entre compresión y calidad puede ser controlada gracias a la definición de un parámetro que determina el rango máximo de las nuevas RAF. Los resultados muestran que es posible lograr factores de compresión de hasta 2.43 con un coeficiente de correlación mayor que 0.9. Esto representa una disminución del volumen de una 58% comparado con el holograma original. Esperamos que estudios subsiguientes permitan lograr mayores compresiones optimizando el algoritmo de generación de RAF, y combinando la técnica presentada con otras técnicas de compresión con pérdida.

REFERENCIAS

1. S. Trejos, J. F. Barrera, A. Velez, M. Tebaldi, and R. Torroba, "Optical approach for the efficient data volume handling in experimentally encrypted data," *J. Opt.* **18**, 065702 (2016).
2. A. Velez, J. F. Barrera, S. Trejos, M. Tebaldi, and R. Torroba, "Optical field data compression by opto-digital means," *J. Opt.* (2016).
3. S. Trejos, J. F. B. Ramirez, A. V. Zea, M. Tebaldi, and R. Torroba, "Compression of multiple 3D color scenes with experimental recording and reconstruction," *Opt. Lasers Eng.* **110**, (2018).

4. R. Shahnaz, J. F. Walkup, and T. F. Krile, "Image compression in signal-dependent noise.," *Appl. Opt.* **38**, 5560–7 (1999).
5. R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik (Stuttg)*. **35**, 237–246 (1972).
6. O. Matoba, T. J. Naughton, Y. Frauel, N. Bertaux, and B. Javidi, "Real-time three-dimensional object reconstruction by use of a phase-encoded digital hologram.," *Appl. Opt.* **41**, 6187–6192 (2002).
7. J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Trans. Inf. Theory* **23**, 337–343 (1977).
8. H. Kadono, H. Takei, and S. Toyooka, "A noise-immune method of phase unwrapping in speckle interferometry," *Opt. Lasers Eng.* **26**, 151–164 (1997).
9. J. R. Buckland, J. M. Huntley, and S. R. Turner, "Unwrapping noisy phase maps by use of a minimum-cost-matching algorithm.," *Appl. Opt.* **34**, 5100–5108 (1995).
10. T. A. Welch, "A Technique for High-Performance Data Compression," *Computer (Long Beach, Calif)*. **17**, 8–19 (1984).
11. D. A. Huffman, "A Method for the Construction of Minimum-Redundance Codes," *Proc. I.R.E.* **40**, 1098–1101 (1952).
12. A. H. Robinson and C. Cherry, "Results of a Prototype Television Bandwidth Compression Scheme," *Proc. IEEE* **55**, 356–364 (1967).