Image processing and computing for digital holography with ImageJ

Procesamiento y cómputo de imágenes para holografía digital con ImageJ

Raúl Castañeda, Pablo Piedrahita-Quintero, Jorge Garcia-Sucerquia*
Universidad Nacional de Colombia-Sede Medellín. Escuela de Física - A.A: 3840-Medellin-050034-Colombia

(*) E-mail: jgarcia@unal.edu.co

Received / Recibido: 18/04/2015

Accepted / Aceptado: 04/06/2015

DOI: 10.7149/OPA.48.2.77

ABSTRACT:

The development of a platform within the framework of ImageJ to process digitally recorded holograms is presented in this work. ImageJ an open source software for processing digital images, provides the needed architecture to develop customized and specialized processing tools of images. In this paper, we show the use of that architecture to develop the needed tools to reconstruct numerically holograms that were digitally recorded. The main advantage of this development is the possibility of using the built-in functions of ImageJ to pre-process the recorded holograms as well as to visualize and manage the reconstructed images. The use of the developed tool is illustrated by means of a step-by-step reconstruction of a digital hologram of a regular die.

Key words: Image processing, ImageJ, plugins, digital holography.

RESUMEN:

Se presenta en este trabajo el desarrollo de una plataforma en el entorno de ImageJ para procesar hologramas registrados digitalmente. ImageJ, un software libre para procesar imágenes digitalmente, ofrece la arquitectura necesaria para desarrollar métodos de procesamiento de imágenes personalizados y especializados. En este artículo mostramos el uso de dicha arquitectura para desarrollar las herramientas necesarias para reconstruir numéricamente hologramas registrados digitalmente. La ventaja principal de este desarrollo es la posibilidad de usar funciones preestablecidas de ImageJ para pre-procesar el holograma registrado, visualizar y manipular las imágenes reconstruidas. El uso de la herramienta desarrollada es ilustrado por medio de la reconstrucción paso a paso de un holograma de un dado común.

Palabras clave: Procesamiento de imágenes, ImageJ, plugin, holografía digital.

REFERENCES AND LINKS / REFERENCIAS Y ENLACES

- [1] G. Dougherty, *Medical image processing: techniques and applications*, Springer Science & Business Media, (2011).
<http://dx.doi.org/10.1007/978-1-4419-9779-1>
- [2] T.C. Poon, T. Kim, "Optical image recognition of three-dimensional objects", *Appl. Opt.* **38**, 370–381 (1999).
<http://dx.doi.org/10.1364/AO.38.000370>
- [3] Z. Gácsi, "The application of digital image processing to materials science", *Mater. Sci. Forum* **414**, 213–220 (2002).
- [4] I. Glaser, H. H. Barrett, "Halftone screen techniques for photographic film response shaping with application to optical processing of medical x-ray images", *Appl. Opt.* **18**, 2294–2300 (1979).
<http://dx.doi.org/10.1364/AO.18.002294>

- [5] S. Kumar, G. Ho, K. M. Woo, L. Zhuo, "Achieving cellular resolution for in vivo retinal images of transgenic GFAP-GFP mice via image processing.", *Opt. Express* **16**, 8250–8262 (2008).
<http://dx.doi.org/10.1364/OE.16.008250>
- [6] T. J. Cornwell, "Image Processing in Radio Astronomy", *Integr. Comput. Imaging Syst.* JW2 (2001).
- [7] J. D. Frost, J. R. Wright, *Digital Image Processing, American Society of Civil Engineers*, New York, NY, USA, (1993).
- [8] <http://www.adcis.net/>
- [9] C. A. Schneider, W. S. Rasband, K. W. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis", *Nat Meth* **9**, 671–675 (2012).
<http://dx.doi.org/10.1038/nmeth.2089>
- [10] M. D. Abràmoff, P. J. Magalhães, S. J. Ram, "Image processing with ImageJ", *Biophotonics Int.* **11**, 36–41 (2004).
- [11] <http://www.nih.gov/>
- [12] T. Ferreira, W. Rasband, *ImageJ User Guide*, (2012).
- [13] W. Burger, M. J. Burge, *Digital image processing: an algorithmic introduction using Java*, Springer Science & Business Media, (2009).
- [14] K. Sierra, B. Bates, *Head first java*, O'Reilly Media, Inc., Sebastopol, CA, (2005).
- [15] <http://rsb.info.nih.gov/ij/plugins/>
- [16] W. Bailer, *Writing ImageJ Plugins—A Tutorial*, (2006).
- [17] U. Schnars, W. P. O. Jüptner, "Digital recording and numerical reconstruction of holograms", *Meas. Sci. Technol.* **13**, 85–101 (2002).
<http://dx.doi.org/10.1088/0957-0233/13/9/201>
- [18] O. K. Ersoy, *Diffraction, Fourier optics and imaging*, John Wiley & Sons, (2006).
- [19] T. Kreis, *Handbook of Holographic Interferometry: Optical and Digital Methods*, Wiley-vch Verlag Ed, Weinheim, (2005).
- [20] U. Schnars, W. Jueptner, *Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*, Springer, Berlin, (2005).
- [21] T. Kreis, *Handbook of Holographic Interferometry: Optical and Digital Methods*, Wiley-VCH, Weinheim; Great Britain, (2004).
<http://dx.doi.org/10.1002/3527604154>
- [22] P. Picart, J. Leval, "General theoretical formulation of image formation in digital Fresnel holography", *J. Opt. Soc. Am. A* **25**, 1744–1761 (2008).
<http://dx.doi.org/10.1364/JOSAA.25.001744>
- [23] E. Cuhe, P. Marquet, C. Depeursinge, "Spatial filtering for zero-order and twin-image elimination in digital off-axis holography", *Appl. Opt.* **39**, 4070–4075 (2000).
<http://dx.doi.org/10.1364/AO.39.004070>

1. Introducción

El procesamiento y cómputo de imágenes digitales son herramientas altamente utilizadas en diferentes ámbitos de la ciencia y la tecnología, como medicina [1], óptica [2], ciencia de los materiales [3], radiología [4], biología [5], astronomía [6] e ingeniería [7]. No obstante debido a las múltiples herramientas que presenta y su facilidad de uso, el software también es utilizado para realizar procesamiento sobre imágenes digitales por usuarios que no necesariamente se desenvuelven en un entorno académico. Actualmente existen diferentes tipos de software que permiten procesar y analizar imágenes digitales. Para la realización de tareas rutinarias en el procesamiento de imágenes, como por ejemplo mejora de contraste, ajuste de brillo, corrección de color detección de bordes, entre otras. Existe un amplio número de paquetes comerciales y de software libre que brindan al usuario un extenso conjunto de herramientas

que permiten realizar tareas como las arriba mencionadas [8]. Sin embargo, cuando se requiere la realización de procesamiento más especializados y/o personalizados, la primera opción de la mayoría de los usuarios es el desarrollo de herramientas propias en plataformas de lenguaje técnico computacional de alto nivel, como MATLAB®, Mathematica®, IDL® o LabView™, debido al relativo corto tiempo de desarrollo cuando se compara con lenguajes como C o C++. Sin embargo esta opción presenta la desventaja que dichos ambientes de desarrollo están orientados algunas veces a un sistema operativo en particular y por lo general no son software de distribución libre, lo que implica que su uso sea restrictivo a un conjunto de usuarios en particular. Otro inconveniente de elegir ambientes de programación en lenguajes como los anteriormente mencionados, es la necesidad de tener que desarrollar las herramientas típicas de las cuales dispone un software comercial para procesar imágenes digitales, demandando su implementación desde cero. Además, si se considera el tiempo invertido para la realización de la GUI (Graphical User Interface), nos encontramos con un panorama desalentador a la hora de utilizar estas alternativas con el fin de personalizar una herramienta que permita procesar imágenes digitales. La anterior situación invita a la búsqueda de un ambiente de desarrollo de tareas personalizadas y especializadas de procesamiento y cómputo de imágenes digitales que se soporte en una plataforma base que provea las herramientas típicas, que opere en la mayoría de los sistemas operativos disponibles y que además sea un software libre (Free Software). Estas características las ofrece el software de procesamiento y análisis de imágenes ImageJ [9,10].

ImageJ permite procesar y computar imágenes digitales, operando sobre cualquiera sistema operativo en la modalidad de dominio público. Este software ofrece todas las herramientas típicas de un software comercial para operar con imágenes y adicionalmente dispone de un entorno optimizado para la realización de tareas personalizadas y específicas de procesamiento y cómputo de imágenes digitales. Dicho entorno está representado en la posibilidad de escribir plugins en el lenguaje de programación Java™, entendidos como aplicaciones que se relacionan con un software principal con el fin de aportarle nuevas funciones. Dichos plugins pueden además hacer uso de las herramientas ofrecidas por ImageJ por medio de simples líneas de comando, lo que deriva en un reducido tiempo de desarrollo enmarcado en un ambiente completo de procesamiento de imágenes digitales.

En este trabajo se introduce ImageJ como una opción válida para llevar a cabo tareas personalizadas y específicas de procesamiento y cómputo de imágenes digitales. De forma particular se muestra el desarrollo de las herramientas necesarias para reconstruir numéricamente hologramas que fueron registrados digitalmente. Se describe brevemente la estructura interna de ImageJ con una marcada intención de orientar al lector a la escritura de plugins. Se ilustra el plugin desarrollado para reconstruir hologramas digitales por medio de la reconstrucción paso a paso de un holograma de un dado. Se finaliza con algunas conclusiones.

2. ImageJ

ImageJ es un software de dominio público que permite realizar procesamiento de imágenes digitales. Desarrollado por el National Institute of Health (NIH) [11], ImageJ puede ser instalado y ejecutado en cualquier ordenador personal con una versión de Java™ igual o superior a 1.1. Se encuentra disponible para diferentes sistemas operativos: Windows, Mac OS y Linux. ImageJ permite editar, analizar, procesar y computar imágenes de 8-bit, 16-bit y 32-bit, en diferentes formatos como GIF, BMP, TIFF, JPEG, DICOM, FITS, entre otros [12]. También cuenta con un conjunto extendido de herramientas de procesamiento de imágenes que permite por ejemplo, la creación de histogramas, manipulación del contraste y la nitidez, detección de bordes, modificar el tamaño y orientación de una imagen, migrar las imágenes en diferentes formatos, entre muchas otras posibilidades. En adición a las herramientas comunes a todos los programas de procesamiento de imágenes, ImageJ cuenta con un conjunto de funciones que permiten analizar imágenes, mediante operaciones matemáticas y/o lógicas, y que pueden utilizarse sobre toda la imagen o regiones específicas de la misma. En la Fig. 1 se muestra el menú principal de ImageJ.

Fig. 1. Menú principal de ImageJ

La ventaja fundamental de ImageJ para el desarrollo de herramientas especializadas y personalizadas es que está diseñado bajo una arquitectura abierta que proporciona la posibilidad de realizar tareas de procesamiento de imágenes digitales mediante macros y plugins. Esta posibilidad permite al usuario resolver casi cualquier problema que involucre el procesamiento, cómputo y análisis de imágenes digitales aprovechando la totalidad de la arquitectura de ImageJ. La estructura interna de ImageJ consiste en un sistema central que es soportado por el lenguaje de programación Java™. Éste se apoya en un kit de herramientas, principalmente de la librería AWT (Abstract Windowing Toolkit), la cual permite realizar de manera cómoda y simple interfaces y sistemas de ventana que funcionan de manera independiente al sistema operativo en el cual se haya desarrollado la aplicación. Dentro de la arquitectura de ImageJ, se encuentran los plugins, que son pequeñas Clases Java™ [13], los cuales se encargan de extender la funcionalidad básica de ImageJ haciendo uso integrado de las herramientas predeterminadas y permitiendo al usuario el desarrollo de nuevas utilidades que personalicen y especialicen el uso de este software. La versatilidad de ImageJ lo convierte en un software especial y atractivo para la realización de procesamiento personalizado y especializado de imágenes. Esta característica única será la utilizada para desarrollar el ambiente requerido para reconstruir numéricamente hologramas registrados digitalmente.

2.1 Plugins en ImageJ

Un plugin es un complemento de software que añade una característica o un servicio específico a un sistema principal. Los plugins construidos en ImageJ se implementan como Clases Java™ [14], esto significa que se pueden utilizar todas las características del lenguaje de programación Java™, lo cual abre un conjunto de posibilidades al momento de desarrollar aplicaciones. Los plugins permiten ser editados y modificados en cualquier momento, además pueden incorporarse al menú principal de ImageJ y se pueden agrupar según su utilidad. ImageJ cuenta con una gran variedad de plugins que han sido realizados por diferentes usuarios alrededor del mundo. Una gran variedad de plugins se pueden encontrar en la página que para este fin ha provisto el NIH [15] y pueden ser descargados de manera gratuita, para su posterior instalación y utilización. Por su característica Open Source, en algunos casos el código fuente de los plugin está disponible para el usuario y se puede modificar con el fin de hacer variantes que puedan ser útiles en situaciones específicas. Ésta es otra de las versatilidades que presenta ImageJ que lo postulan como una herramienta adecuada para tratamiento personalizado y especializado de imágenes digitales, como es el caso en la reconstrucción numérica de hologramas registrados digitalmente.

Existen diferentes formas de escritura de plugins en ImageJ de acuerdo a las necesidades del usuario [16]. Si éste desea generar una aplicación que no necesite trabajar directamente sobre una imagen abierta, se recomienda utilizar el tipo *plugin*. Por el contrario si se requiere un algoritmo que realice operaciones sobre imágenes que han sido previamente abiertas lo más recomendable es que se utilice el tipo *plugin filter*. Por último si se necesita una interfaz gráfica, sobre la cual el usuario ingresa o selecciona parámetros necesarios para llevar a cabo un procesamiento, se recomienda el tipo *plugin frame*. Los detalles necesarios para la escritura de un plugin de cualquiera de los tipos se pueden leer en [16].

3. Reconstrucción de hologramas digitales en ImageJ

Con el fin de aprovechar el entorno que ofrece ImageJ para procesar imágenes digitales en la reconstrucción numérica de hologramas registrados digitalmente, se desarrolló un plugin del tipo *filter* en el cual se implementó el método de transformada de Fresnel, los detalles de la implementación numérica de esta transformada pueden leerse en [17–19]. El software y su código fuente están disponibles al público en general en la dirección <http://unal-optodigital.github.io/NumericalPropagation/> bajo la sección de descargas.

La holografía digital [20] se puede entender como un proceso de formación de imágenes de dos pasos. El primero consiste en el registro del holograma digital, el cual es el patrón de interferencia que se registra de la superposición de una onda objeto y una onda de referencia sobre la superficie de una cámara digital; ésta puede ser tipo CCD o CMOS. La onda objeto proviene del objeto que se desea registrar y la onda de referencia incide directamente desde la fuente de iluminación en la cámara digital; el patrón de interferencia es una imagen digital que se almacena en la memoria de una computadora y que se denomina *holograma digital* $I_{n,m}$. En la Fig. 2 se esquematiza el proceso de registro de un holograma digital, detalles adicionales de este proceso se pueden leer en [21].

Fig. 2 Esquema para representar el proceso de registro de un holograma digital. DHV: divisor de haz variable, OM: objetivo de microscopio, L: lente, E1, E2: espejos, HO: haz objeto, HR: haz de referencia, CDH: cubo divisor de haz, CMOS: cámara digital.

El segundo paso es realizar la reconstrucción numérica del holograma digital, haciendo uso de algoritmos de propagación de campos ópticos [18]. Básicamente la reconstrucción consiste en simular el fenómeno de difracción que sufre la onda de referencia al incidir sobre el holograma digital. En el plugin desarrollado este proceso de difracción es modelado por medio de la transformada de Fresnel [17]:

$$U_{p,q}(z) = \frac{\exp(ikz)}{i\lambda z} \exp\left(\frac{ik}{2z} \Delta \xi^2 p^2\right) \exp\left(\frac{ik}{2z} \Delta \eta^2 q^2\right) \sum_{n=-N/2}^{(N/2)-1} \sum_{m=-M/2}^{(M/2)-1} I_{n,m} \exp\left(\frac{ik}{2z} (\Delta x^2 n^2 + \Delta y^2 m^2)\right) \exp\left(-\frac{ik}{z} (\Delta x n \Delta \xi p + \Delta y m \Delta \eta q)\right) \quad (1)$$

En la ecuación anterior se considera que la onda de referencia es plana e incide perpendicularmente sobre el holograma digital $I_{n,m}$, el cual fue registrado por una cámara digital con $N \times M$ píxeles de dimensiones $\Delta x, \Delta y$. El número de píxeles de la cámara, el plugin lo extrae automáticamente de la imagen abierta y activa en ImageJ sobre la cual se calculará la ecuación (1). El tamaño asignado a la imagen, en dimensiones físicas, es introducido por el usuario en la ventana principal del plugin. De este tamaño se extraen las dimensiones de los píxeles Δx y Δy . Una imagen típica de un holograma digital se presenta en la Fig. 3, para la cual las dimensiones físicas serán introducidas en la ventana principal, por ejemplo este holograma digital, denominado *105.tif* tiene 768 x 576 píxeles y dimensiones 8.45 mm x 6.34 mm.

Fig. 3. Holograma digital abierto y activo en ImageJ.

La ventana principal del plugin se muestra en la Fig. 4. Para recrear numéricamente el proceso físico de la difracción representado por la ecuación (1), el plugin requiere parámetros físicos de entrada que hacen posible realizar la reconstrucción del holograma. Estos parámetros son: *Lambda*: la longitud de onda de la fuente de luz monocromática utilizada para realizar la reconstrucción del holograma digital. *Distance of propagation*: distancia en la cual se evaluará el proceso de difracción; si la onda de referencia del proceso de reconstrucción es idéntica a la utilizada en el registro, esta distancia debe ser igual a la cual se encontraba el objeto de la cámara digital. *Screen size X* y *Screen size Y*: dimensiones físicas del sensor

utilizado como medio de registro en los ejes X , Y , respectivamente. Estos parámetros son ingresados por medio del cuadro de diálogo de la Fig. 4.

Fig. 4. Ventana principal del plugin DH_OD

El paso inicial de proceso de reconstrucción del holograma digital se realiza por medio del filtrado espacial. El entorno del plugin realiza el filtrado espacial por medio del botón *FFT* de la Fig. 4. Una vez pulsado este botón se abre una nueva ventana que corresponde al espectro de la transformada de Fourier del holograma digital como se muestra la Fig. 5 (a) y emerge el mensaje mostrado en la Fig. 5 (b). Con el fin de mejorar la visualización de la transformada de Fourier, su espectro es visualizado en escala logarítmica para optimizar el uso del rango dinámico de la imagen. De esta forma los tres órdenes difractados que componen el holograma [22] pueden ser representados en una única imagen a pesar que el orden cero es al menos mil veces más intenso que los órdenes ± 1 . El proceso de reconstrucción debe separar la imagen real o la imagen virtual de los otros dos órdenes para propagar el patrón de difracción de la imagen seleccionada. Esta tarea de extracción del orden de interés, conocida como filtrado espacial [23], se realiza por la selección de las frecuencias espaciales correspondientes a través de una de las herramientas de selección provistas por ImageJ. El mensaje emergente mostrado en la Fig. 5 (b), invita al usuario a realizar dicha selección.

Fig. 5. Transformada de Fourier del holograma digital (a) y mensaje emergente para seleccionar región de filtrado espacial (b).

Una vez el usuario selecciona las frecuencias espaciales correspondientes a la imagen que desea propagar, se debe pulsar el botón *Propagation*, con lo cual el plugin calculará la ecuación (1) usando los parámetros introducidos en la ventana principal sobre el conjunto de frecuencias espaciales seleccionadas. El resultado de este cálculo brinda la amplitud de campo complejo $U_{p,q}(z)$ a partir de la cual podemos calcular la intensidad:

$$I_{p,q}(z) = U_{p,q}(z) U_{p,q}^*(z) \quad (2)$$

o la fase

$$\phi_{p,q}(z) = a \tan \frac{\text{Im}[U_{p,q}(z)]}{\text{Re}[U_{p,q}(z)]}. \quad (3)$$

El resultado de la propagación es desplegado en una nueva imagen presentada en una ventana emergente. Para realizar esta tarea el plugin automáticamente ajusta el contraste y brillo del holograma reconstruido para visualizar la imagen con el mejor uso posible del rango dinámico. Para ilustrar las imágenes resultantes del proceso de reconstrucción numérica de hologramas registrados digitalmente con el uso del plugin desarrollado, se presentan en la Fig. 6 (a) y (b) las imágenes reconstruidas en amplitud cuando se seleccionan diferentes regiones del espectro de Fourier del holograma de la Fig. 3; el espectro es el ilustrado en la Fig. 5 (a). En la Fig. 6(a) se han seleccionado todas las frecuencias espaciales que componen el espectro transformada de Fourier del holograma digital. Esta selección permite la propagación de todos los constituyentes del holograma digital: i) las imágenes real y virtual del objeto, correspondientes a la imagen del dado y a la nube de puntos brillantes y oscuros, respectivamente y ii) el orden cero de difracción que corresponde al recuadro brillante en el centro de la Fig. 6(a). Cuando la selección de la región de interés en el espacio de las frecuencias se limita a aquellas que representan la imagen real, se logra eliminar el orden cero y la imagen virtual, lo cual conduce a un holograma reconstruido como el que se presenta en la Fig. 6 (b).

La incorporación del plugin para reconstruir numéricamente hologramas registrados digitalmente en el entorno de ImageJ, ofrece la ventaja de disponer de todo el conjunto de herramientas para procesamiento y análisis que ofrece este software libre. Sobre la imagen de salida se puede por ejemplo realizar: detección de bordes, manipulación del histograma, almacenamiento en diferentes formatos, pseudocoloreado, medición de distancias y ángulos, entre muchas más tareas. Este plugin puede además ser extendido para incorporar otros métodos de propagación numérica de campos ópticos y de esta forma ampliar su espectro de uso.

Fig. 6. Imágenes reconstruidas para diferentes selecciones de filtrado espacial. (a) Se han seleccionado todas las frecuencias espaciales en el espectro de Fourier; (b) Solamente las frecuencias correspondientes a la imagen real se han seleccionado para el proceso de propagación

4. Conclusiones

En este trabajo se ha presentado el uso del software libre de procesamiento de imágenes ImageJ para el análisis, mejoramiento y reconstrucción numérica de hologramas registrados digitalmente. Puesto que el proceso de reconstrucción del holograma digital se puede entender como el cálculo numérico de la difracción que sufre una onda electromagnética en el holograma registrado, es decir se requiere del cómputo de la integral de difracción, la primera opción para realizar dicha tarea es realizarla software de programación de alto nivel, como Matlab®, Mathematica®, IDL®, LabView™, entre otros. Sin embargo cuando el tiempo de cómputo debe reducirse, se recurre a lenguajes como C, C++ o Fortran. En cualquiera de las dos opciones, adicionalmente a la elaboración de la herramienta de reconstrucción del holograma

se impone la programación de herramientas adicionales para el mejoramiento del holograma registrado, análisis de los resultados, etc.

En este trabajo se ha presentado el uso de la plataforma ImageJ, el cual es un software de distribución libre altamente versátil para procesar y editar imágenes digitales, para el desarrollo de las herramientas necesarias para procesar y reconstruir numéricamente hologramas registrados digitalmente. El uso de la plataforma ofrecida por ImageJ para el desarrollo de herramientas especializadas, permite la integración de los algoritmos propios de la holografía digital en el ambiente más amplio y robusto de procesamiento de imágenes que ofrece este software libre. Esta integración facilita el pre-procesamiento de los hologramas registrados, la visualización de los hologramas reconstruidos, el análisis de los resultados obtenidos, entre otras tareas, por medio del uso de sólidas herramientas desarrolladas y probadas para el entorno de ImageJ.

Se ha ilustrado la herramienta desarrollada por medio del procesamiento y reconstrucción de un holograma digital de un dado. Se ha ilustrado el proceso de reconstrucción visualizando los diferentes órdenes de difracción, el filtrado espacial digital para finalizar con la presentación mejorada del holograma reconstruido.

Agradecimientos

Los autores agradecen el soporte de la Universidad Nacional de Colombia, Proyecto Hermes 19384, el Programa de Internacionalización del Conocimiento y el programa de Jóvenes Investigadores de Colciencias-Universidad Nacional de Colombia código Hermes 28751.