

Fast phase processing in off-axis holography using multiplexing with complex encoding and live-cell fluctuation map calculation in real-time

Pinhas Girshovitz^{1,2} and Natan T. Shaked^{1,*}

¹Department of Biomedical Engineering, Faculty of Engineering, Tel Aviv University, Tel Aviv 69978, Israel

²pinhasgi@tau.ac.il

nshaked@tau.ac.il

Abstract: We present efficient algorithms for rapid reconstruction of quantitative phase maps from off-axis digital holograms. The new algorithms are aimed at speeding up the conventional Fourier-based algorithm. By implementing the new algorithms on a standard personal computer, while using only a single-core processing unit, we were able to reconstruct the unwrapped phase maps from one megapixel off-axis holograms at frame rates of up to 45 frames per second (fps). When phase unwrapping is not required, the same algorithms allow frame rates of up to 150 fps for one megapixel off-axis holograms. In addition to obtaining real-time quantitative visualization of the sample, the increased frame rate allows integrating additional calculations as a part of the reconstruction process, providing sample-related information that was not available in real time until now. We use these new capabilities to extract, for the first time to our knowledge, the dynamic fluctuation maps of red blood cells at frame rate of 31 fps for one megapixel holograms.

©2015 Optical Society of America

OCIS codes: (090.1995) Digital holography; (090.5694) Real-time holography; (100.5070) Phase retrieval; (100.5088) Phase unwrapping; (110.3175) Interferometric imaging; (120.3180) Interferometry.

References and links

1. P. Girshovitz and N. T. Shaked, "Generalized cell morphological parameters based on interferometric phase microscopy and their application to cell life cycle characterization," *Biomed. Opt. Express* **3**(8), 1757–1773 (2012).
2. N. Turko, I. Barnea, O. Blum, R. Korenstein, and N. T. Shaked, "Detection and controlled depletion of cancer cells using photothermal phase microscopy," accepted to *J. Biophoton.* (2014).
3. N. Shoham, P. Girshovitz, R. Katzung, N. T. Shaked, D. Benayahu, and A. Gefen, "Adipocyte stiffness increases with accumulation of lipid droplets," *Biophys. J.* **106**(6), 1421–1431 (2014).
4. B. Kemper, A. Bauwens, A. Vollmer, S. Ketelhut, P. Langehanenberg, J. Muthing, H. Karch, and G. von Bally, "Label-free quantitative cell division monitoring of endothelial cells by digital holographic microscopy," *J. Biomed. Opt.* **15**(3), 036009 (2010).
5. N. T. Shaked, L. L. Satterwhite, N. Bursac, and A. Wax, "Whole-cell-analysis of live cardiomyocytes using wide-field interferometric phase microscopy," *Biomed. Opt. Express* **1**(2), 706–719 (2010).
6. B. Rappaz, A. Barbul, Y. Emery, R. Korenstein, C. Depeursinge, P. J. Magistretti, and P. Marquet, "Comparative study of human erythrocytes by digital holographic microscopy, confocal microscopy, and impedance volume analyzer," *Cytometry A* **73A**(10), 895–903 (2008).
7. S. Grilli, V. Vespini, F. Merola, P. Ferraro, L. Miccio, M. Paturzo, and S. Coppola, "Exploring the capabilities of digital holography as tool for testing optical microstructures," *3D Res.* **2**, 1 (2011).
8. B. Kemper, P. Langehanenberg, and G. von Bally, "Digital holographic microscopy: a new method for surface analysis and marker-free dynamic life cell imaging," *Optik and Photonik* **2**(2), 41–44 (2007).
9. S. Gawad, M. Giugliano, M. Heuschkel, B. Wessling, H. Markram, U. Schnakenberg, P. Renaud, and H. Morgan, "Substrate arrays of iridium oxide microelectrodes for in vitro neuronal interfacing," *Front. Neuroeng.* **2**, 1 (2009).
10. C. Edwards, A. Arbabi, G. Popescu, and L. L. Goddard, "Optically monitoring and controlling nanoscale topography during semiconductor etching," *Light: Sci. Appl.* **30**, 1–6 (2012).

11. H. Pham, H. Ding, N. Sobh, M. Do, S. Patel, and G. Popescu, "Off-axis quantitative phase imaging processing using CUDA: toward real-time applications," *Biomed. Opt. Express* **2**(7), 1781–1793 (2011).
12. P. Girshovitz and N. T. Shaked, "Real-time quantitative phase reconstruction in off-axis digital holography using multiplexing," *Opt. Lett.* **39**(8), 2262–2265 (2014).
13. S. K. Debnath and Y. K. Park, "Real-time quantitative phase imaging with a spatial phase-shifting algorithm," *Opt. Lett.* **36**(23), 4677–4679 (2011).
14. B. Bhaduri and G. Popescu, "Derivative method for phase retrieval in off-axis quantitative phase imaging," *Opt. Lett.* **37**(11), 1868–1870 (2012).
15. Y. Xu, Y. Wang, W. Jin, C. Lv, and H. Wu, "A new method of phase derivative extracting for off-axis quantitative phase imaging," *Opt. Commun.* **305**, 13–16 (2013).
16. B. Sha, X. Liu, X. L. Ge, and C. S. Guo, "Fast reconstruction of off-axis digital holograms based on digital spatial multiplexing," *Opt. Express* **22**(19), 23066–23072 (2014).
17. M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J. Opt. Soc. Am. A* **72**(1), 156–160 (1982).
18. P. Girshovitz and N. T. Shaked, "Compact and portable low-coherence interferometer with off-axis geometry for quantitative phase microscopy and nanoscopy," *Opt. Express* **21**(5), 5701–5714 (2013).
19. M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," *Appl. Opt.* **41**(35), 7437–7444 (2002).
20. I. Shock, A. Barbul, P. Girshovitz, U. Nevo, R. Korenstein, and N. T. Shaked, "Optical phase nanoscopy in red blood cells using low-coherence spectroscopy," *J. Biomed. Opt.* **17**(10), 101509 (2012).
21. N. T. Shaked, L. L. Satterwhite, M. J. Telen, G. A. Truskey, and A. Wax, "Quantitative microscopy and nanoscopy of sickle red blood cells performed by wide field digital interferometry," *J. Biomed. Opt.* **16**(3), 030506 (2011).
22. G. Popescu, Y. K. Park, W. Choi, R. R. Dasari, M. S. Feld, and K. Badizadegan, "Imaging red blood cell dynamics by quantitative phase microscopy," *Blood Cells Mol. Dis.* **41**(1), 10–16 (2008).
23. Y. K. Park, M. Diez-Silva, G. Popescu, G. Lykotrafitis, W. Choi, M. S. Feld, and S. Suresh, "Refractive index maps and membrane dynamics of human red blood cells parasitized by *Plasmodium falciparum*," *Proc. Natl. Acad. Sci. USA* **105**(37), 13730–13735 (2008).
24. K. Kim, H. Yoon, M. Diez-Silva, M. Dao, R. R. Dasari, and Y. K. Park, "High-resolution three-dimensional imaging of red blood cells parasitized by *Plasmodium falciparum* and in situ hemozoin crystals using optical diffraction tomography," *J. Biomed. Opt.* **19**(1), 011005 (2014).
25. Y. Jang, J. Jang, and Y. K. Park, "Dynamic spectroscopic phase microscopy for quantifying hemoglobin concentration and dynamic membrane fluctuation in red blood cells," *Opt. Express* **20**(9), 9673–9681 (2012).

1. Introduction

Off-axis digital holography is an imaging method for capturing the complex wave-front (amplitude and phase) of an imaged object by encoding it into an interference pattern acquired by a digital camera. The ability to capture the complex wave-front has made digital holography attractive in many fields, including label-free imaging for biological applications [1–6] and nondestructive testing in metrology [7–10].

Off-axis interference in digital holography enables wave-front acquisition at the camera frame rate, without multiple frames per each sample instance, making it suitable for dynamic imaging. However, the extraction of the recorded sample wave-front is computationally intense, and thus it is usually done offline. This digital extraction process typically includes spatial filtering by two 2-D Fourier transforms, and 2-D phase unwrapping to solve ambiguities in the phase map of the sample, which yields the unwrapped phase map. Typically, when using a conventional computer and utilizing a single processing core, this full digital process (spatial filtering and phase unwrapping) can take half a second for one megapixel holograms, which precludes video-frame processing and visualization [11,12].

Currently, to enable inline processing of off-axis holograms, one can use the graphic processing unit (GPU) of the computer that can divide the overall calculation to smaller calculations, each is performed on one of the GPU multiple internal processors in parallel [11], while speeding up the total calculation time. However, GPU processing requires special programming skills, does not work on all types of computers, and does not decrease the inherent computational complexity of the algorithms. Another solution is using significantly different wave-front extraction algorithms with less heavy computational needs [13–15]. However, these algorithms typically induce certain limitations on the samples that can be quantitatively imaged.

We lately showed that by algorithmically improving the conventional off-axis holographic processing method, which is based on two 2-D Fourier transforms and 2-D phase unwrapping, it is possible to reach video frame rates, while utilizing only a single-core processing unit of a standard personal computer [12]. These algorithms are capable of reaching unwrapped-phase-map processing frame rates of up to 32 fps for one megapixel off-axis holograms. This is done by significantly decreasing the number of pixels used in the phase unwrapping step, and by improving the efficiency of the Fourier transforms through decreasing the number of calculations performed per processed hologram. In that work, we showed that by using digital multiplexing, two off-axis holograms with orthogonal fringe directions can be processed by a single Fourier transform. Sha, et al. [16] suggested an enhanced version of our previous paper, and digitally multiplexed four holograms that are processed by a single Fourier transform, under the assumption of a uniaxial fringe direction.

In the current paper, we present more efficient algorithms for wave-front extraction and phase reconstruction in off-axis holography. Those algorithms significantly improve the efficiency of the Fourier transform steps and enable higher processing frame rates. The new algorithms are capable of reaching processing frame rates of up to 45 fps for one megapixel off-axis holograms when unwrapped phase maps are needed, and frame rates of up to 150 fps for one megapixel off-axis holograms when the phase unwrapping is not required.

Using this new approach, we have processed the phase maps of a sample of red blood cells (RBCs) on a regular computer, and in the first time to our knowledge, showed processing of the cell membrane fluctuations maps at video frame rate for one megapixel off-axis holograms. In addition, we combined fast calculation of the thickness maps of the RBCs and their maps of membrane fluctuations for two megapixels holograms.

2. Reconstruction algorithms for phase retrievals from off-axis holograms

The recorded interference pattern on the digital camera, also called the off-axis hologram, can be mathematically expressed by the following equation:

$$\begin{aligned} I(x, y) &= |E_s(x, y)|^2 + |E_r|^2 + E_s^*(x, y)E_r + E_s(x, y)E_r^* \\ &= I_s(x, y) + I_r + 2\sqrt{I_s(x, y)I_r} \cos\left(\frac{2\pi}{\lambda} \left[OPD(x, y) - y \sin(\theta_y) - x \sin(\theta_x) \right] \right), \end{aligned} \quad (1)$$

where E_s and E_r are the sample and reference complex waves, respectively, I_s and I_r are the intensities of sample and reference waves, respectively, λ is the illumination wavelength, OPD is the total optical path delay (or optical thickness) of the sample, and θ_x and θ_y are the off-axis angles between the sample and reference waves in relation to the x and y axes, respectively, assuming straight fringes. For a well-designed system, by controlling θ_x and θ_y , a full separation between the auto-correlation terms (E_s^2 and E_r^2 in the spatial-frequency domain) and the cross-correlation terms ($E_s^*E_r$ and $E_sE_r^*$ in the spatial-frequency domain) is obtained, which allows complete extraction of the sample wave-front from Eq. (1).

We next review the conventional algorithm (Section 2.1), the two previously suggested algorithms (Sections 2.2 and 2.3) [12], and then present three new and more efficient algorithms (Sections 2.4-2.6), for quantitative phase extraction from off-axis holograms.

2.1 Algorithm A: The conventional algorithm

The reconstruction of the sample quantitative phase map is usually performed by using the conventional algorithm, termed here as Algorithm A, which is based on spatial filtering using Fourier transforms [17]. This algorithm, presented in Fig. 1, includes the following steps:

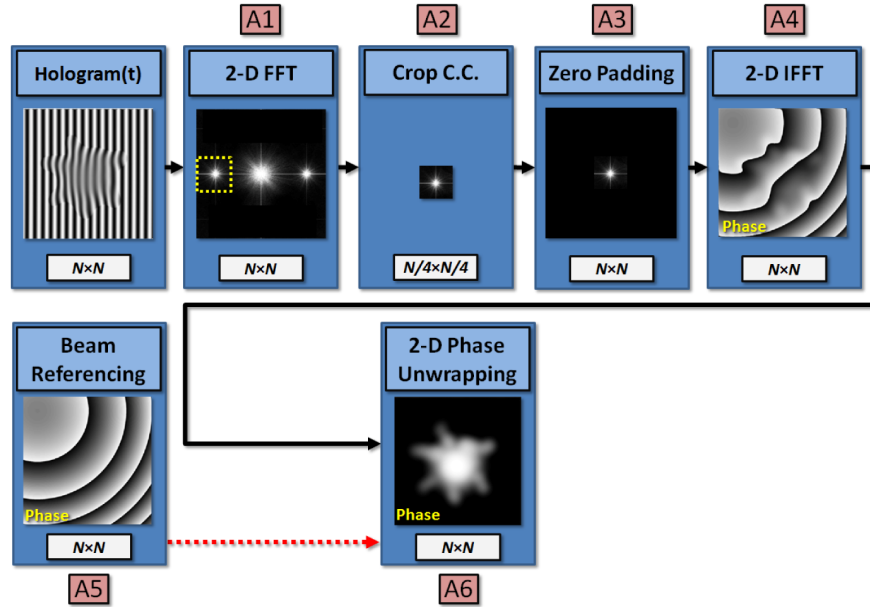


Fig. 1. Algorithm A: The conventional algorithm for quantitative phase map reconstruction from off-axis holograms.

- A1. Two-dimensional fast Fourier transform (2-D FFT): Convert the digital hologram, containing $N \times N$ real pixels, to the spatial-frequency domain using a 2-D FFT, resulting in a matrix containing $N \times N$ complex pixels.
- A2. Cross-correlation cropping: Crop the $N/4 \times N/4$ cross-correlation (marked by the yellow square in Fig. 1). In an efficient system, the entire wave-front spatial-frequency content occupies $N/4 \times N/4$ pixels.
- A3. Zero padding: Insert the cropped cross-correlation to the center of an empty matrix containing $N \times N$ pixels.
- A4. 2-D inverse FFT (2-D IFFT): Convert the zero-padded cross-correlation back to the image domain using a 2-D IFFT, resulting in an $N \times N$ complex matrix representing the sample wave-front.
- A5. Beam referencing: To compensate for stationary aberrations and curvatures in the beam profile, before the experiment, acquire a hologram without the sample, and process it into the sample-free wave-front using steps A1-A4 elaborated above. Then, divide the sample wave-front from step A4 by the sample-free wave-front.
- A6. Phase unwrapping: The argument of the resulting $N \times N$ complex matrix is the wrapped phase map of the sample, and it is subjected to 2π ambiguities in cases that the optical thickness of the sample is greater than the illumination wavelength. In these cases, we apply a 2-D phase unwrapping algorithm, and obtain an $N \times N$ matrix representing the unwrapped phase map, free of 2π ambiguities.

2.2 Algorithms B: The cropped cross-correlations algorithm

In [12], Girshovitz et al. presented Algorithm B that decreases the number of pixels calculated by the 2-D IFFT and by the unwrapping algorithm. The main improvement is simply based on skipping step A3, and thus reducing the matrix size for the followed steps by a factor of 16, without losing sample information. This is possible since the image in the initial off-axis hologram is spatially sampled at a higher rate to avoid overlaps between the cross-correlation

and the auto-correlation terms. Algorithm B includes the following steps, and its diagram is given in [12]:

- B1. 2-D FFT: Same as step A1.
- B2. Cross-correlation cropping: Same as step A2.
- B3. 2-D IFFT: Convert the $N/4 \times N/4$ cropped cross-correlation back to the image domain by using a 2-D IFFT, resulting in an $N/4 \times N/4$ complex matrix representing the sample wave-front.
- B4. Beam referencing: Before the experiment, calculate the $N/4 \times N/4$ sample-free wave-front using steps B1-B3 above, and divide the sample wave-front by the sample-free wave-front.
- B5. Phase unwrapping: The argument of the resulting $N/4 \times N/4$ matrix is the wrapped phase map. To solve 2π ambiguities, apply a 2-D phase unwrapping, which results in the $N/4 \times N/4$ unwrapped phase matrix.
- B6. Enlarge unwrapped phase map: Enlarge the $N/4 \times N/4$ unwrapped phase matrix to the final $N \times N$ unwrapped phase matrix.

2.3 Algorithm C: The hologram multiplexing algorithm

The next algorithm, also presented in [12] and termed as Algorithm C, includes digital multiplexing of two off-axis holograms from the hologram sequence. This allows using a single Fourier transform on a multiplexed hologram, rather than two Fourier transforms, which further decreases the overall calculation time. Algorithm C includes the following steps, and its diagram is given in [12]:

- C1. Hologram multiplexing: Sum hologram I_1 with a transverse (90° -rotated) version of the next hologram $T\{I_2\}$, which yields the multiplexed hologram $I_A = I_1 + T\{I_2\}$ containing $N \times N$ pixels.
- C2. 2-D FFT: Same as step A1 or B1.
- C3. Cross-correlation cropping $\times 2$: Crop the vertical cross-correlation and the horizontal cross-correlation, each containing $N/4 \times N/4$ pixels.
- C4. 2-D IFFT $\times 2$: Same as step B3, but for both the horizontal and the vertical cross-correlation terms. This results in two $N/4 \times N/4$ complex wave-fronts.
- C5. Transposing: Rotate the complex wave-front originated from the vertical cross-correlation in -90° .
- C6. Beam referencing $\times 2$: Same as step B4, but for both wave-fronts.
- C7. Phase unwrapping $\times 2$: Same as step B5, but for both wave-fronts.
- C8. Enlarge unwrapped phase map $\times 2$: Same as step B6, but for both wave-fronts.

Next, we present three new algorithms, termed here as Algorithms D-F, that allow significant processing-time improvements compared to the algorithms presented before.

2.4 Algorithm D: The complex FFT algorithm

Since Fourier transform is an operation performed on complex variables and the recorded holograms do not have an imaginary part, the typically-used Fourier transform calculation is inefficient. As such, it is possible to exploit the imaginary part to create a complex hologram from two successive holograms. Each of these initial holograms may be a multiplexed hologram, containing two wave-fronts, as presented in Algorithm C. Then, by using a single 2-D FFT on the complex multiplexed hologram, we process four wave-fronts simultaneously. After the 2-D FFT, since the real and imaginary parts are mixed, decoding is needed to extract the four original cross-correlation terms.

The encoding of the complex hologram I_C is done as follows:

$$I_C(m, n) = I_A(m, n) + jI_B(m, n), \quad (2)$$

where I_A and I_B are two multiplexed holograms, and n and m are the discrete coordinates of the holograms. Thus, in the first stage, we encode the two successive multiplexed holograms I_A and I_B into a single complex hologram I_C .

For extraction of the four wave-fronts encoded into the complex multiplexed hologram I_C , only a single 2-D FFT on I_C is needed. To show this, let us look at the inverse connections between the complex hologram and its real and imaginary parts:

$$I_A(m, n) = \frac{I_C(m, n) + I_C^*(m, n)}{2}; I_B(m, n) = j \frac{I_C^*(m, n) - I_C(m, n)}{2}. \quad (3)$$

Then, we note that $FFT\{I_C^*\}(m, n) = FFT\{I_C\}^*(M - m, N - n) = Flip\{FFT\{I_C\}^*(m, n)\}$, where N and M are the sizes of the matrix. Therefore, the 2-D FFT of I_C^* can be calculated out of the 2-D FFT of I_C by flipping it on the two axes. As such, the 2-D FFT of I_A and I_B can be calculated only from the 2-D FFT of I_C , and can be expressed as follows:

$$FFT\{I_A\}(m, n) = \frac{FFT\{I_C\}(m, n) + Flip\{FFT\{I_C\}^*(m, n)\}}{2};$$

$$FFT\{I_B\}(m, n) = j \frac{Flip\{FFT\{I_C\}^*(m, n)\} - FFT\{I_C\}(m, n)}{2}. \quad (4)$$

Therefore, in order to calculate the 2-D FFT of I_A and I_B , it is enough to calculate only once the 2-D FFT of I_C and then use Eq. (4).

Furthermore, since only one cross-correlation term from each conjugate pair is useful for us, there is no need to perform the addition and subtraction operations of Eq. (4) on the entire matrices. Instead, note that the flipping operation simply puts the +1 cross-correlation term on the -1 cross-correlation term. Therefore, Eq. (4) can be implemented only on the relevant cross-correlation terms. This is demonstrated in Fig. 2, presenting Algorithm D, which includes the following steps:

- D1. Hologram multiplexing: Sum hologram I_1 with a transverse (90°-rotated) version of the next hologram $T\{I_2\}$, which yields the multiplexed hologram $I_A = I_1 + T\{I_2\}$ containing $N \times N$ pixels. Do the same for the next two holograms I_3 and I_4 , to yield another complex hologram $I_B = I_3 + T\{I_4\}$.
- D2. Complex summation: Sum the two multiplexed holograms I_A and I_B as a complex matrix, where the first multiplexed hologram is the real part of this matrix and the second multiplexed hologram is its imaginary part. This creates a complex hologram I_C according to Eq. (2), containing $N \times N$ pixels.
- D3. 2-D FFT: Convert the complex digital hologram I_C , containing $N \times N$ real pixels, to the spatial-frequency domain using a 2-D FFT, resulting in a matrix containing $N \times N$ complex pixels.
- D4. Cross-correlation cropping $\times 4$: Crop the two vertical cross-correlation terms and the two horizontal cross-correlation terms, each containing $N/4 \times N/4$ pixels.
- D5. Cross-correlation extraction $\times 4$: Eq. (4) is performed only on the required cross-correlation terms (each of them contains $N/4 \times N/4$ pixels), which yields the horizontal and vertical cross-correlation terms for I_A and the horizontal and vertical cross-correlation terms for I_B .

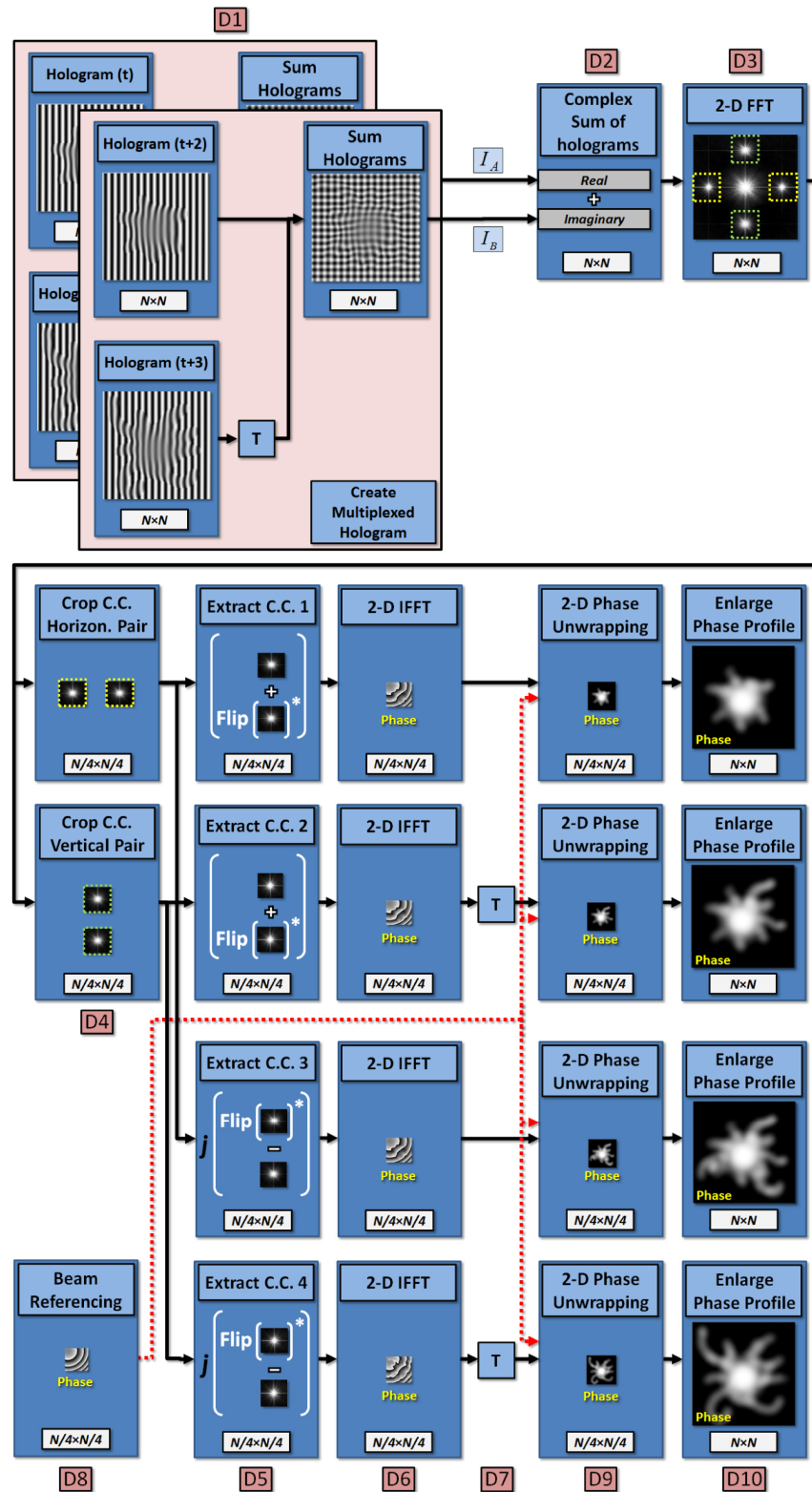


Fig. 2. Algorithm D: The complex FFT algorithm.

- D6. 2-D IFFT $\times 4$: Convert the $N/4 \times N/4$ extracted horizontal and vertical cross-correlation terms back to the image domain by using 2-D IFFTs, resulting in four $N/4 \times N/4$ complex matrices, representing the sample wave-fronts.
- D7. Transposing $\times 2$: Same as C5, but for the two rotated wave-fronts.
- D8. Beam referencing $\times 4$: Same as step B4, but for the four wave-fronts.
- D9. Phase unwrapping $\times 4$: Same as step B5, but for the four wave-fronts.
- D10. Enlarge unwrapped phase profile $\times 4$: Same as step B6, but for the four wave-fronts.

2.5 Algorithm E: The re-sampling algorithm

In Algorithm C [12] and Algorithm D, we exploited empty areas in the spatial-frequency domain for inserting additional cross-correlation terms, and process all of them with the same 2-D FFT. Still, there are more unused areas in the spatial-frequency domain that can be used. Typically, the sampling rate is equal on both axis, but only one axis of the off-axis hologram contains the fast carrier frequency for the off-axis fringes, which is required for preventing aliasing between the auto-correlation and the cross-correlation terms at one direction. For this reason, it is possible to re-sample down the hologram on the other axis, without losing sample information. Therefore, in Algorithm E we assume that the off-axis fringes are on one of the axes, so that either θ_x or θ_y from Eq. (1) are equal to zero. Then, asymmetric re-sampling of the recorded hologram by a factor of 4 is performed, only across the axis on which there is no off-axis angle. This re-sampling does not damage the other axis, in which the off-axis angle creates the cross-correlation terms separation from the auto-correlation terms, which enables full extraction of the cross-correlation terms. Then, 1-D FFT can be used instead of 2-D FFT, which is done orthogonally to the fringe direction. For the general case, where neither θ_x or θ_y are equal to zero, a diagonal re-sampling is required.

Let us assume that the interference fringes are straight across vertical axis. Algorithm E is presented in Fig. 3, and includes the following steps:

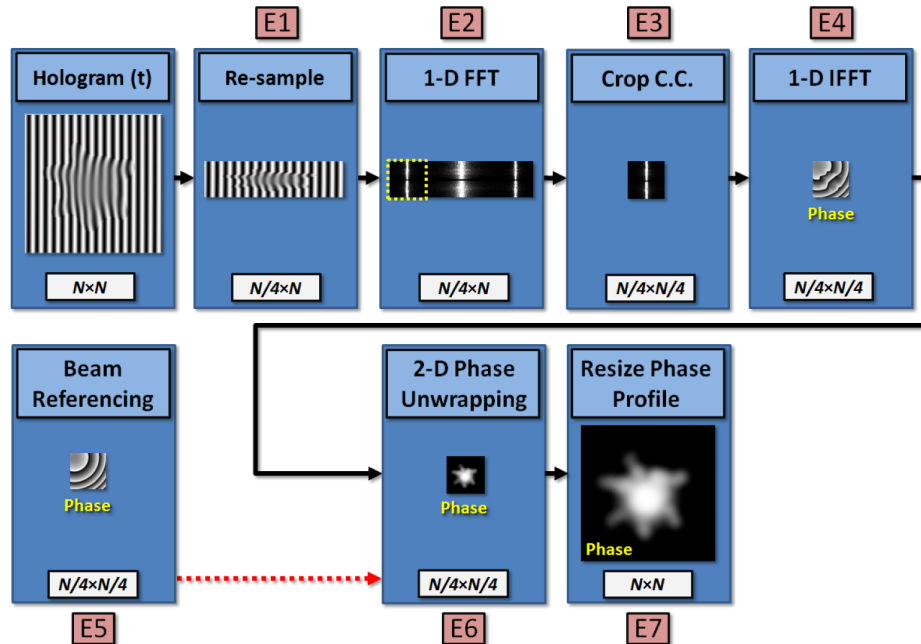


Fig. 3. Algorithm E: The re-sampling algorithm.

- E1. Re-sampling: Re-sample the $N \times N$ digital hologram along the vertical axis to create a hologram with $N/4 \times N$ pixels.
- E2. 1-D FFT: Convert the digital hologram, containing $N/4 \times N$ pixels, to the spatial-frequency domain using a 1-D FFT across the horizontal axis, resulting in a matrix containing $N/4 \times N$ complex pixels.
- E3. Cross-correlation cropping: Same as step A2.
- E4. 1-D IFFT: Convert the $N/4 \times N/4$ cross-correlation term back to the image domain using a 1-D IFFT across the horizontal axis, resulting in an $N/4 \times N/4$ complex matrix representing the sample wave-front.
- E5. Beam referencing: Before the experiment, calculate the $N/4 \times N/4$ sample-free wave-front using steps E1-E4 elaborated above, and divide the sample wave-front by the sample-free wave-front.
- E6. Phase unwrapping: Same as step B5.
- E7. Enlarge unwrapped phase profile: Same as step B6.

2.6 Algorithm F: The complex FFT re-sampling algorithm

Final improvement can be done to Algorithm E by implementing the same analysis of the complex hologram defined in Eqs. (2)-(4), but this time without using the hologram multiplexing stage.

Thus, in this case, we will not use multiplexed holograms in Eq. (2) (but rather complex holograms with fringes on one direction only), and the flip operation of Eq. (4) is defined only around the horizontal axis (in the same direction of the 1-D FFT).

Let us assume that the interference fringes are straight across vertical axis. Algorithm F is presented in Fig. 4, and includes the following steps:

- F1. Re-sampling $\times 2$: Re-sample the two $N \times N$ digital holograms I_1 and I_2 along the vertical axis to create a hologram with $N/4 \times N$ pixels.
- F2. Complex summation: Sum the two resampled holograms as a complex matrix where the first hologram is the real part and the second hologram is the imaginary part, to create a complex hologram I_C according to Eq. (2) (where in our case I_A is the first re-sampled hologram and I_B is the second re-sampled hologram, each of which contains $N/4 \times N$ pixels).
- F3. 1-D FFT: Same as step E2.
- F4. Cross-correlation cropping $\times 2$: Crop the two cross-correlation terms, each containing $N/4 \times N/4$ pixels.
- F5. Cross-correlation extraction $\times 2$: Eq. (4) is performed only on the cropped cross-correlation terms with a one-directional flipping, which yields the cross-correlation term for I_A and the cross-correlation term for I_B .
- F6. 1-D inverse FFT $\times 2$: Same as step E4, but for the two extracted cross-correlation terms.
- F7. Beam referencing $\times 2$: Same as step E5, but for the two wave-fronts.
- F8. Phase unwrapping $\times 2$: Same as step B5, but for the two phase profiles.
- F9. Enlarge unwrapped phase profile $\times 2$: Same as E7 for the two phase profiles.

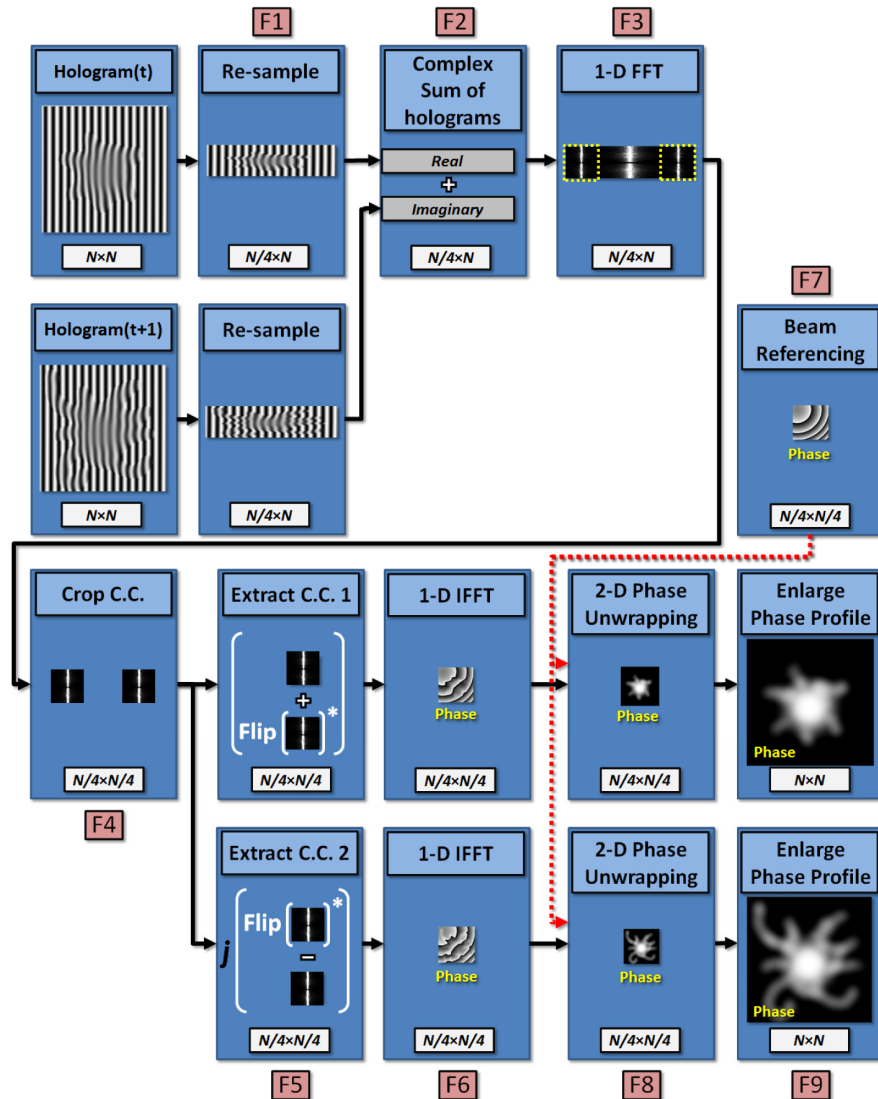


Fig. 4. Algorithm F: The complex FFT re-sampling algorithm.

3. Comparison between the algorithms

To evaluate the performance of the algorithms, off-axis image holograms were acquired using a portable interferometric module connected to a transmission microscope [18]. The digital processing was done using a conventional personal computer (Intel i7-2600, 3.4GHz CPU, 8GB RAM), without using GPU or parallel processing (only a single core was utilized), on Matlab R2012b. The phase unwrapping was carried out using the 2D-SRNCP algorithm [19].

The first stage in the evaluation of the algorithms included measuring the frame rates with and without phase unwrapping, when processing off-axis holograms of various sizes. The second part of the evaluation included measuring the calculation times of each of the different steps of the algorithms for an off-axis hologram containing one megapixel.

For measuring the frame rates, we used five data sets of 400 off-axis holograms containing 2048×2048 , 1024×1024 , 768×768 , and 512×512 pixels each. Each evaluated parameter was determined based on an averaged value of five runs of each algorithm and for

each data set. Figure 5 shows the frame rates achieved using the different algorithms, per hologram size, with phase unwrapping (Fig. 5(a)) and without it (Fig. 5(b)).

Table 1 presents a comparison of the processing times of the different stages in the algorithms for one-megapixel holograms. Compared to Algorithm A, the newly presented Algorithms D-F suggest much faster calculation times for the FFT and the IFFT steps. Additional decrease in the calculation time can be seen in the phase unwrapping step, due to the exclusion of the zero-padding step [12].

In comparison to Algorithm A, Algorithm D shows a decrease by a factor of 6 in the calculation time of the 2-D FFT, since it calculates four wave-fronts with a single Fourier transform of a complex hologram composed of two multiplexed holograms, in addition to other FFT-related calculations that are now done only once. A decrease by a factor of 10 is seen in the calculation time of the inverse 2-D FFT, due to the smaller calculated matrix (can also be seen by comparing Algorithms B and C [12]).

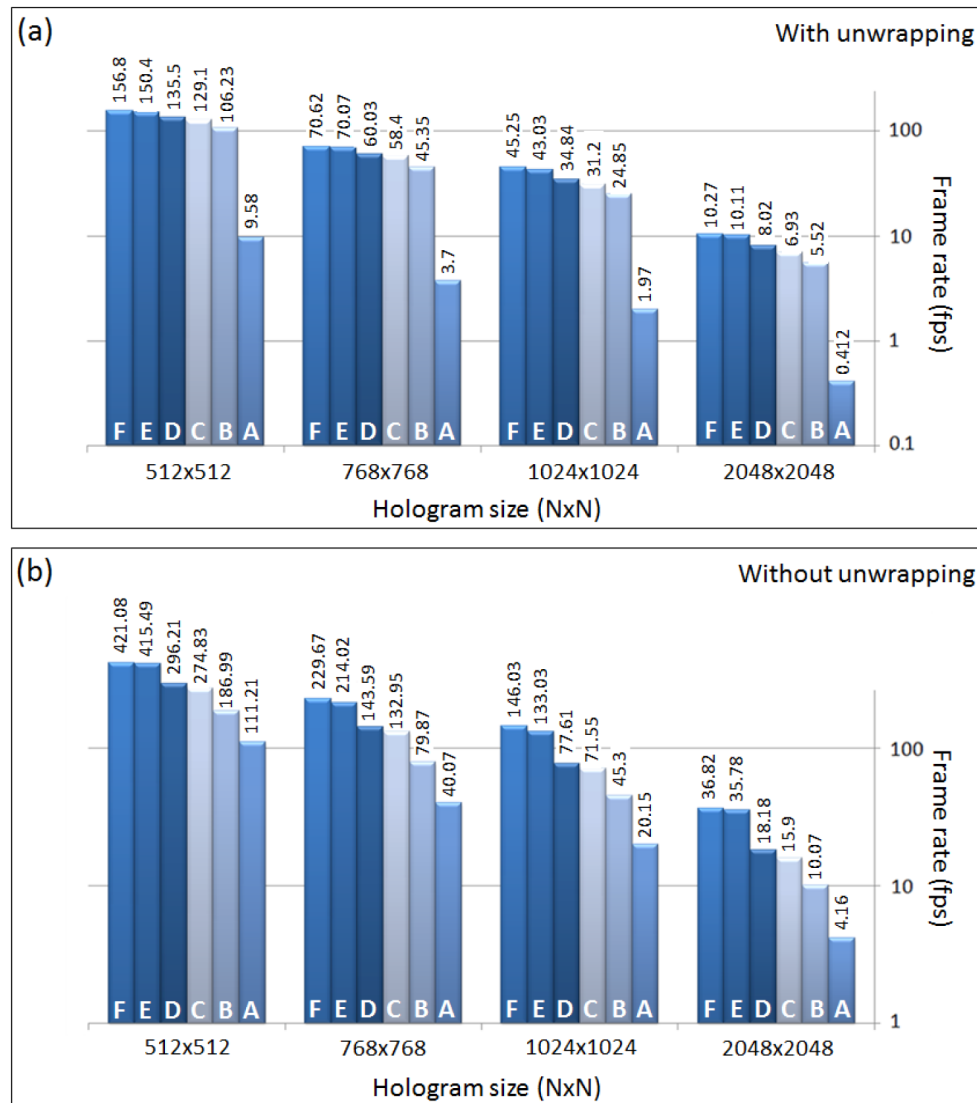


Fig. 5. Comparison between the frame rates (in fps) of Algorithms A-F for various sizes of off-axis holograms. (a) When using phase unwrapping. (b) Without using phase unwrapping.

Table 1. Comparison between the processing times and frame rates of the various algorithms for off-axis holograms containing 1024×1024 pixels.

Algorithm		Time [ms]					Frame rate [fps]	
		FFT	IFFT	Unwrap	Others	Total±SD	With unwrapping	Without unwrapping
Previously presented	A	13.81	13.44	457.97	22.38	507±20	1.97	20.15
	B	14.00	1.44	17.13	7.63	40.2±0.16	24.85	45.30
	C	6.58	1.38	17.13	5.00	32.1±0.25	31.2	71.55
Newly presented	D	2.30	1.32	15.71	9.38	28.7±0.32	34.84	77.61
	E	0.63	0.33	15.82	6.46	23.23±0.11	43.04	133.03
	F	0.42	0.33	15.77	5.58	22.10±0.14	45.25	146.03

In Algorithms E and F, a different approach is applied. We re-sampled the hologram on one axis, and applied a 1-D FFT on the other axis. This enables us to decrease the calculation time by a factor of 22 for Algorithm E and by a factor of 33 for Algorithm F. Overall, the use of a 1-D FFT caused an additional reduction in the calculation time of the IFFT by a factor of 4 compared to Algorithm D, and a factor of 40.5 compare to Algorithm A.

In [12], Girshovitz et al. have experimentally validated that there is no resolution decrease when processing holograms by Algorithms B and C, in comparison to the conventional Algorithm A. To confirm the quality of the reconstruction for the newly presented Algorithms D-F, a 1951 USAF phase test target, created by focused ion beam lithography, was measured. The recorded hologram was processed to the optical thickness map using Algorithms A, and D-F. From Fig. 6(a), we can see that the four algorithms show similar results, with the same reconstructed resolution limit of 0.69 μm (group 9, element 4 in the USAF test target). This experimental resolution limit corresponds to the calculated resolution limit of the imaging system of 0.7 μm . Figure 6(b) shows cross-sections across the optical thickness maps at group 8, between the points indicated by the white arrows in Fig. 6(a), for the four reconstructions. We can see that the four algorithms provide similar results, with minor artefacts, smaller than the point spread function of the imaging system, which are related to numerical noise.

To evaluate the noise robustness of the new algorithms, a Gaussian white noise was added to the initial hologram, creating an SNR of 12.4 dB. Then, the optical thickness maps were extracted using all six algorithms. Next, we compared the results in the same manner carried out in Fig. 6, and visibility obtained the exact same profiles for all algorithms (data is not shown). Furthermore, the calculated SNR in all optical thickness maps from all six algorithms was about 9.5 dB. We therefore conclude that all new five algorithms (B-F) have the same noise robustness as the conventional algorithm (A).

4. Real-time calculation of the thickness and fluctuation maps of red blood cells

The high frame rates provided by the new algorithms enable additional in-process calculations, while still maintaining video frame rate for visualization of the unwrapped phase maps. To demonstrate this, we used the IPM setup to image RBC samples. We acquired 500 off-axis holograms containing 1024×1024 pixels and 1024×2048 pixels, at recording frame rates of 31 fps and 15 fps, respectively. Then, we applied algorithm F for the extraction of the unwrapped phase maps of the sample. Since for RBCs, the refractive index can be considered as homogenous, the physical thickness map $h(x, y; t)$ of the sample can be derived from the

time-dependent optical thickness map $OPD(x, y; t)$ by dividing it by $\Delta n = 0.065$ ($n_{cell} = 1.395$ and $n_{media} = 1.33$) [20].

During this physical thickness map calculation process, we have integrated an additional calculation for the temporal fluctuations in the thickness map, which is associated with the root mean square (RMS) membrane displacement of the RBCs, a parameter that was previously shown useful for characterizing blood-related diseases [21–23], and is defined as follows:

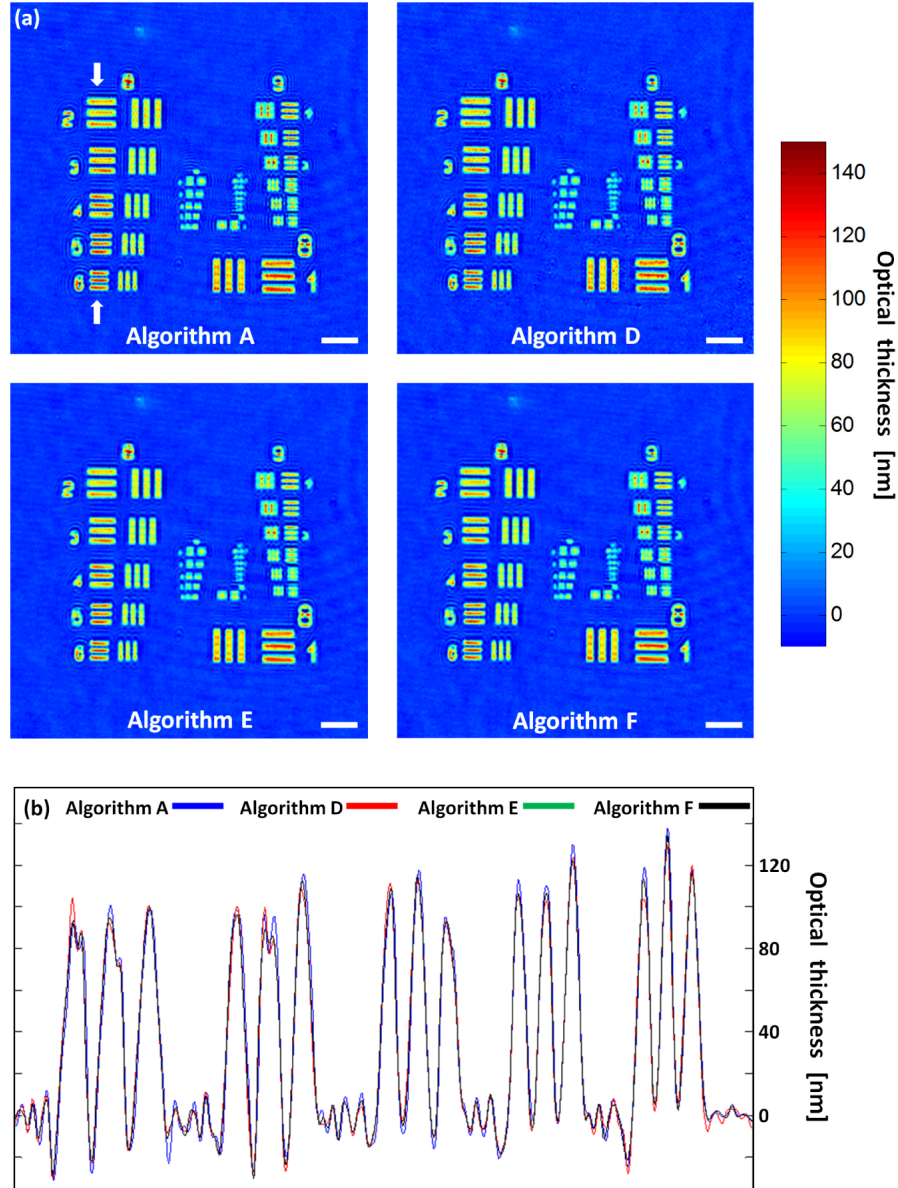


Fig. 6. (a) Resolution comparison of the optical thickness maps between the conventional algorithm (Algorithm A) and the newly-presented fast algorithms (Algorithms D, E, and F), showing that in spite of significant speedup in the processing time, the quality of the reconstructions of the new algorithms was not damaged. White scale bars represent 10 μm upon the sample. (b) Graphs presenting the vertical cross sections between the white arrows shown in (a).

$$\Delta h_{RMS}(x, y) = \sqrt{\left\langle \left(h(x, y; t) - \langle h(x, y; t) \rangle_t \right)^2 \right\rangle_t}, \quad (5)$$

where $h(x, y; t)$ is the physical thickness map of the sample at time point t , and $\langle \bullet \rangle_t$ is the temporal average for each pixel (x, y) . Thus, to calculate the RMS membrane displacement map $\Delta h_{RMS}(x, y)$, for each spatial point (x, y) on the thickness map, we need a temporal vector of points. For this aim, during the run of Algorithm F, between steps F8 and F9, a stack was integrated to store up to 24 temporal frames, which change over time in a 'first in – first out' (FIFO) stack manner. Finally, the calculated RMS membrane displacement map should be pushed into step F9 for enlargement and presentation.

Figure 7(a) shows one frame from this thickness map sequence. [Media 1](#) and Fig. 7(b) present the resulting RMS membrane displacement map of the RBCs, containing 1024×1024 pixels. Using this approach, we obtained processing frame rate of 31 fps (and presented at a frame rate of 25 fps due to visualization time consumption). To the best of our knowledge, this is the first time where the RMS membrane displacement map is calculated dynamically at real time. This map reveals interesting information about the dynamics of the fluctuations of the RBCs. From [Media 1](#), we can see that the fluctuations amplitude create wave-like pattern, which either rotates around the RBC or is localized on a small area on the cell surface.

In our second demonstration, we processed off-axis holograms, which are double in size compared to the ones used in the first demonstration. This time, we integrated the calculation of the RMS membrane displacement map for a 1024×256 -pixel window, shifted across the field of view, during the visualization for the thickness map, containing 1024×2048 pixels. The result can be seen in Fig. 8 and in [Media 2](#). The overall processing frame rate was 22 fps without the fluctuation-map calculation, and 15 fps with the fluctuation-map calculation. We believe that this new technique will find perspective uses for real-time cell diagnosis and sorting.

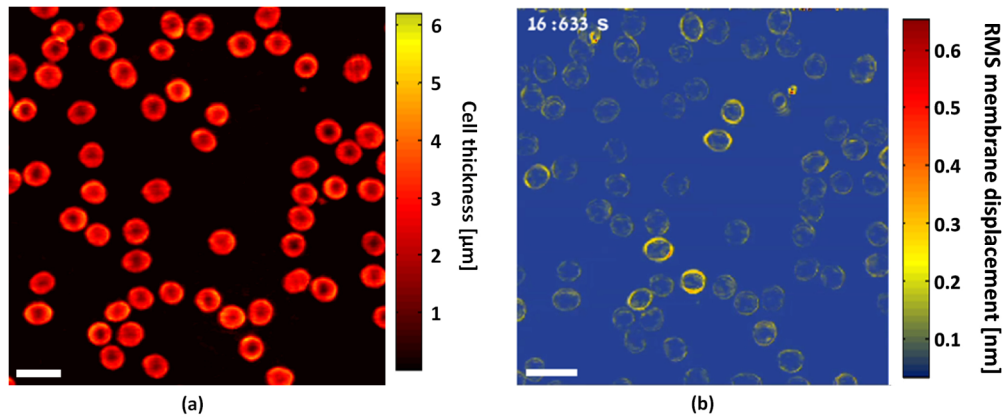


Fig. 7. (a) Quantitative physical thickness maps of RBCs, obtained from an off-axis hologram containing 1024×1024 pixels. (b) RMS membrane displacement map of the RBCs, obtained by applying Algorithm F for the dynamically-changing thickness map, and using a FIFO stack of 24 points in time for each pixel. The calculation frame rate is 31 fps (see full dynamics in [Media 1](#)). All calculations were performed on a single-core processing unit of a conventional personal computer. White scale bars represent 10 μm .

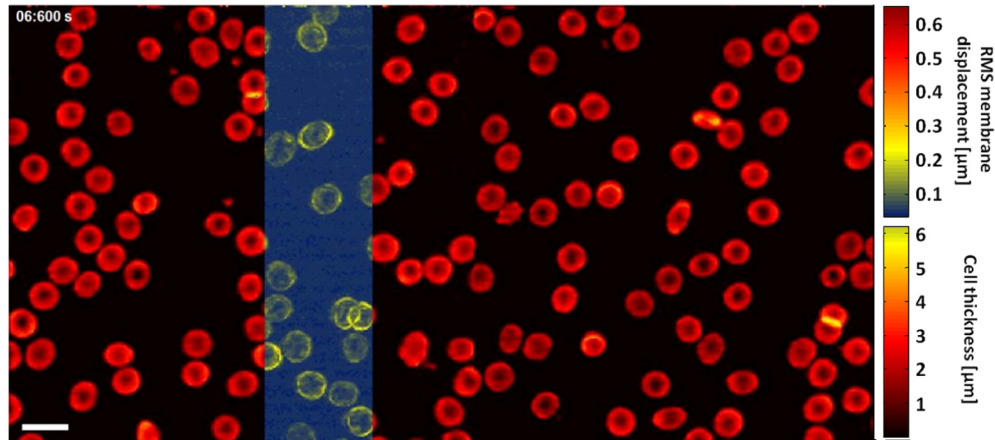


Fig. 8. Dynamic quantitative physical thickness maps of RBCs, obtained by applying Algorithm F for off-axis holograms containing 1024×2048 pixels. At the same time of the phase map calculation, we have calculated the dynamic RMS membrane fluctuation map in a window of 1024×256 pixels, shifted across the field of view, as obtained by a FIFO stack of 24 points in time per each pixel. The overall frame rate of these two calculations performed together is 15 fps (see full dynamics in [Media 2](#)). All calculations were performed on a single-core processing unit of a conventional personal computer. White scale bar represents $10 \mu\text{m}$ upon the sample.

5. Conclusions

We presented new and efficient algorithms for quantitative phase map reconstruction, reaching processing frame rates of up to 45 fps for one megapixel off-axis holograms when the unwrapped phase map is needed, and processing frame rates of up to 150 fps when phase unwrapping is not required, using a single-core processing unit on a standard personal computer. This was done by increasing the efficiency of the Fourier transform steps in the conventional algorithm. We demonstrated that since the phase map reconstruction time can exceed video frame rate, additional sample-related parameters could be calculated, while still maintaining video frame rate.

In general, reaching higher frame rates allows using larger camera sensors, which contain more pixels, for real-time quantitative imaging. Here, we presented that for two megapixel holograms, it is possible to reach near video frame rate, while using phase unwrapping. In addition, we demonstrated that even four-megapixel holograms can still be reconstructed with reasonable frame rate of 10 fps when using phase unwrapping or 36 fps without phase unwrapping. This presents a new standard for performance and enables imaging significantly larger areas on the samples in real time, on standard personal computers.

After comparing the performances of the various algorithms, we demonstrated rapid processing in reconstructing the quantitative unwrapped phase and thickness maps of RBC samples, and simultaneous inline calculation of the dynamic RMS membrane displacement map of the RBCs. This parameter was previously shown as a useful tool for characterization of blood-related diseases, and the ability to visualize it in real time provides a new tool for fast analysis and diagnosis on a higher number of cells together. Furthermore, the proposed method might be useful to significantly speed up the processing time of other interferometric techniques that use multiple Fourier transforms, including tomographic phase microscopy [24] and spectroscopic quantitative phase imaging [25].

Acknowledgments

The research was supported by the FP7 Marie Curie Career Integration Grant No. 303559.