UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

Camila de Paula D'Almeida

Development of lens-free holographic microscopes using multiheight and multispectral phase recovery methods

São Carlos

2023

Camila de Paula D'Almeida

Development of lens-free holographic microscopes using multiheight and multispectral phase recovery methods

Thesis presented to the Graduate Program in Physics at the Instituto de Física de São Carlos da Universidade de São Paulo, to obtain the degree of Doctor in Science.

Concentration area: Theoretical and Experimental Physics

Advisor: Prof. Dr. Sebastião Pratavieira

Corrected version

(Original version available on the Program Unit)

São Carlos

2023

D'Almeida, Camila de Paula
Development of lens-free holographic microscopes using multiheight and multispectral phase recovery methods / Camila de Paula D'Almeida; advisor Sebastião Pratavieira - corrected version -- São Carlos 2023.
155 p.

Thesis (Doctorate - Graduate Program in Theoretical and Experimental Physics) -- Instituto de Física de São Carlos, Universidade de São Paulo - Brasil , 2023.

1. Lensless optical microscopy. 2. Phase recovery. 3. Multiheight. 4. Multispectral. 5. Open-source. I. Pratavieira, Sebastião, advisor. II. Title.

To my friend Rony, who during our brief friendship left a mark that will remain forever in my life.

ACKNOWLEDGEMENTS

Como se sabe, todo um doutorado não cabe somente em uma centena de páginas. Existem vivências e aprendizados que possibilitam a produção científica. Portanto, esta seção foi escrita para que eu pudesse agradecer a todos que estiveram comigo ao longo desses anos, contribuindo de forma direta ou indireta para que esse trabalho fosse concretizado.

Gostaria de agradecer primeiramente a Deus, que foi e será meu sustento diante de todas as situações. Agradeço também à constante intercessão de Nossa Senhora.

Agradeço ao professor Sebastião Pratavieira, que me aceitou como sua primeira aluna de pós-graduação no início do mestrado e tem acompanhado meu trabalho com muito zelo até aqui. Agradeço pelos incentivos e pela confiança que tem em mim.

Agradeço à Natália Portes de Oliveira, ao Patrick Oliveira Feitosa, ao Felipe Alvarenga Carvalho, ao Cesar Yudi Kuramoto e ao Gustavo Fernandes da Costa, alunos de iniciação científica e estágio que se dedicaram para que pudéssemos juntos trabalhar neste projeto.

Agradeço também ao Marlon Rodrigues Garcia, que esteve na supervisão de alguns projetos de iniciação científica, além de auxiliar diretamente nos desafios da instrumentação desenvolvida. Agradeço à Kamila Jessie Sammarro Silva por colaborar nos experimentos com cistos de *Giardia*, executando e me ensinando os procedimentos de análise por imunofluorescência, além de contribuir com discussões e escrita dos resultados. À Camilla Costa dos Santos pelo preparo das culturas de células, pelo auxílio nos ensaios de migração celular e também por contribuir com discussão e escrita dos resultados. À Giulia Kassab e à Carolina de Paula Campos pelo auxílio nos primeiros experimentos com células, preparando as culturas celulares e testando protocolos. Ao Everton Sérgio Estracanholli e à Fernanda Alves pela orientação e auxílio no preparo das amostras de leveduras. E também à Loraine Carolina Goenaga Mafud e Ana Paula da Silva pela assistência no uso dos laboratórios de microbiologia.

Agradeço à professora Lyda Patricia Sabogal Paz, do Departamento de Hidráulica e Saneamento da Escola de Engenharia de São Carlos (EESC), por conceder gentilmente a suspensão de cistos de Giardia e o kit para análise por imunofluorescência. À professora Maria do Carmo Calijuri, do mesmo departamento, pela concessão de amostras de microalgas e à professora Odete Rocha pela concessão de amostras de *Ceriodaphnias*, provenientes do Laboratório de Limnologia e Ecotoxicologia Aquática da Universidade Federal de São Carlos (UFSCar), por intermédio da Francine Perri Venturini. Agradeço ao professor Clovis Wesley Souza, do Laboratório de Microbiologia e Parasitologia da UFSCar, por

também ceder amostras de microalgas e me ensinar gentilmente como prepará-las para observação no microscópio.

Agradeço aos professores do Grupo de Óptica, de forma especial ao professor Vanderlei Salvador Bagnato, à professora Cristina Kurachi e ao professor Francisco Eduardo Gontijo Guimarães.

Agradeço também aos pesquisadores do Grupo de Óptica, de forma especial à Lilian Tan Moriyama, que acompanhou este projeto e dedicou tempo para oferecer valiosas considerações para o aprimoramento deste documento, e ao José Dirceu Vollet Filho pelo constante apoio e pela ajuda na estruturação das ideias.

Agradeço aos funcionários do Laboratório de Instrumentação Eletrônica (LIEPO), em especial ao João Marcelo Pereira Nogueira, aos funcionários do Laboratório de Apoio Técnico (LAT), em especial ao Vinícius Sigari e aos funcionários da Oficina de Óptica, em especial ao Tiago Luis Firmiano. Também agradeço aos funcionários da Biblioteca do Instituto de Física de São Carlos (IFSC) e de todo IFSC que possibilitaram esse trabalho.

Agradeço de forma especial aos meus pais, Milton e Maria Aparecida e aos meus irmãos, Marcos e Matheus, que sempre estiveram comigo e me incentivaram a estudar. Agradeço a toda minha família, que tantas vezes sem compreender o caminho que eu seguia, estava sempre acompanhando meu crescimento, oferecendo muito amor.

Agradeço ao Johan, meu namorado, por seguir crescendo comigo nesses últimos anos, se fazendo presente de forma muito especial na minha vida e me incentivando a ir além.

Agradeço a cada um dos meus amigos e colegas de laboratório, que fizeram essa experiência de pós-graduação ser única e possível. Agradeço aos meus amigos do Ministério Universidades Renovadas (MUR) e a todos que fazem parte da minha vida.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento direto desta pesquisa através da bolsa de doutorado (processo 88882.328745/2019 − 01). Também agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (processos 465360/2014-9 e 306919/2019-2), ao Programa Unificado de Bolsas (PUB) da USP, e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) - processos: 2013/07276-1 (CEPOF), 2014/50857-8 (INCT) e 2019/19793-7 - que contribuem de forma significativa para promoção das pesquisas dentro do Grupo de Óptica "Prof. Dr. Milton Ferreira de Souza", no qual este trabalho foi desenvolvido.

Agradeço por fim, a todos cidadãos brasileiros que contribuem para que a Universidade de São Paulo e tantas outras Universidades públicas se sustentem e ofereçam estudos e pesquisas de qualidade para nosso país.

"Para ser grande, sê inteiro: nada
Teu exagera ou exclui.
Sê todo em cada coisa. Põe quanto és
No mínimo que fazes.
Assim em cada lago a lua toda
Brilha, porque alta vive"
Ricardo Reis

ABSTRACT

D'Almeida, C. P. **Development of lens-free holographic microscopes using multiheight and multispectral phase recovery methods**. 2023. 155p. Tese (Doutorado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Since its invention in century XVI, microscopy is one of the most important optics techniques. Its use has revolutionized many knowledge areas and these applications have driven improvements in the technique itself. The lens-free microscope has emerged as an alternative to traditional optical microscopes. This recent technique has the advantage to decouple the imaging resolution from the field of view and also to provide very compact and low-cost systems. Lens-free microscopes based on hologram acquisition produce amplitude and phase images of the same observed scene, using a digital imaging process to perform phase recovery. This study aims to develop and test portable and automated lens-free holographic microscopes, based on open-source software, using two methods for imaging acquisition and processing: multiheight and multispectral. The first applied method is the multiheight, which records a sequence of holograms with different sensor-sample distances and uses them as input data to recover phase values of the light that better describes the recorded scene. Multispectral method was also used to record holograms using light with emission at different spectral ranges and composes amplitude and phase images from this imaging information. Homemade open-source algorithms were developed in Python language to control the component device and to perform the digital imaging processing. Both lens-free microscopes were tested using standard test targets to define the spatial resolution of each one. Besides, it has been applied to imaging different types of biological samples to analyze their amplitude and phase images in different conditions. Thus, these systems were used to image different cells and microorganisms. More robust analyses were performed to provide yeast and Giardia cyst counting. Therefore, two portable, low-cost, automated, and based on open-source software lens-free holographic microscopes were developed and tested in this study, using multiheight and multispectral methods to perform amplitude and phase images with almost $30mm^2$ and about $5\mu m$ of resolution. Both systems proved to be efficient for observing cells and microorganisms without the need for labels.

Keywords: Lensless optical microscopy. Phase recovery. Multiheight. Multispectral. Open-source.

RESUMO

D'Almeida, C. P. **Desenvolvimento de microscópios holográficos sem lentes usando métodos de recuperação de fase multialtura e multiespectral**. 2023. 155p. Tese (Doutorado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Desde a sua invenção no século XVI, a microscopia é uma das técnicas ópticas mais importantes. Seu uso revolucionou muitas áreas do conhecimento e essas aplicações têm impulsionado melhorias na própria técnica. O microscópio sem lentes surgiu como uma alternativa aos microscópios ópticos tradicionais. Esta técnica recente tem a vantagem de desacoplar a resolução da imagem do campo de visão, além de fornecer sistemas muito compactos e de baixo custo. Os microscópios sem lente baseados na aquisição de holograma produzem imagens de amplitude e fase da mesma cena observada, usando um processamento digital de imagem para realizar a recuperação de fase. Este estudo tem como objetivo desenvolver e testar microscópios holográficos sem lentes portáteis e automatizados, baseados em algoritmos computacionais de códigos abertos, aplicando dois métodos para aquisição e processamento das imagens: multialturas e multiespectral. O primeiro método aplicado é o multialturas, que registra uma sequência de hologramas com diferentes distâncias entre o sensor e a amostra e os utiliza como dados de entrada para recuperar os valores de fase da luz que melhor descrevem a cena gravada. O método multiespectral também foi usado para registrar hologramas, usando luz com emissão em diferentes faixas espectrais para compor imagens de amplitude e fase a partir dessas informações em forma de imagem. Algoritmos de código aberto foram desenvolvidos em linguagem Python para controlar os dispositivos que compõem o sistema e realizar os processamentos digitais das imagens. Ambos os microscópios sem lentes foram testados usando alvo de teste padrão para definir a resolução espacial de cada um. Além disso, eles foram aplicados para observar diferentes tipos de amostras biológicas, de modo a analisar suas imagens de amplitude e fase em condições distintas. Assim, esses microscópios foram usados para observar diferentes tipos de células e microorganismos. Análises mais robustas foram realizadas para fornecer mais informações a respeito das amostras. Portanto, dois microscópios holográficos sem lentes portáteis, de baixo custo, automatizados e baseados em algoritmos computacionais de código aberto foram desenvolvidos e testados neste estudo, usando métodos multialturas e multiespectrais para realizar imagens de amplitude e fase com quase $30mm^2$ e cerca de $5\mu m$ de resolução. Ambos sistemas se mostraram eficientes para observar células e microorganismos sem a necessidade de marcação.

Palavras-chave: Microscópio óptico sem lentes. Recuperação de fase. Multialturas. Multiespectral. Código aberto.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

2D	Two dimensions
3D	Three dimensions
ASM	Angular Spectrum Method
ATCC	American type culture collection
CMOS	Complementary metal-oxide-semiconductor
DMEM	Dulbecco's modified Eagle medium
EDTA	Ethylenediaminetetraacetic acid
HDFn	Human dermal fibroblasts neonatal
HUVEC	Human umbilical vein cells
FBS	Fetal Bovine Serum
FOV	Field of view
FWHM	Full width at half maximum
GUI	Graphical user interface
IFA	Immunofluorescence assay
LED	Light-emitting diode
LOD	Limit of detection
LHM	Lens-free holographic microscopy
LOD	Limit of detection
PBS	Phosphate buffered saline
RGB	Red, green and blue
SCC	Squamous cell carcinoma
TC	Tamura coefficient
USA	United States of America
USAF	United States Air Force

USEPA	United States Environmental Protection Agency
YPD	Yeast extract - peptone - dextrose

LIST OF SYMBOLS

A	Amplitude values of the electrical field of the light
λ	Wavelength
ϕ	Phase values of the electrical field of the light
$\Delta\lambda$	Spectral width
D	Diameter of the pinhole
E	Electric field of a light wave
$\mathcal{F}$	Fourier Transform
$\mathcal{F}^{-1}$	Inverse of the Fourier Transform
k	Wave vector
n	Refractive index
z_1	Distance between the light source and the sample
z_2	Distance between the sample and the digital image sensor
Δz	Step distance for hologram acquisition in the multiheight LHM system
μ	Mean value of cysts concentration
σ	Standard deviation of cysts concentration

CONTENTS

1 INTRODUCTION

Human vision is very efficient in its function of facilitating day-to-day tasks. However, more detailed investigations, driven by the curiosity to understand meticulously what surrounds us, showed the biological limitation of our eyes.

The use of equipment such as optical instruments, still without much theoretical understanding, followed by the improvement of lens manufacturing, allowed human eyes to have access to much smaller details than those provided by their natural capabilities. (1)

In addition to assisting in everyday activities that required more of the human eye, microscopy also contributed to the evolution of scientific understanding. And this, in turn, further encouraged the progress of these optical techniques. (1)

Since the first microscopes, built at the end of the 16th century, the magnification of optical images was based on equipment composed of lenses. Although it is evident that significant advances in the understanding and fabrication of these basic optical elements, it is interesting to note that, even today, lenses are essential components for most optical imaging techniques. While maintaining this feature in common, microscopes were being built quite differently. Initially, their operation was based on the transmission and reflection of light by the samples, phenomena more commonly observed in everyday life. As science and technology progressed, these systems have expanded their performance capacity, providing images by phase contrast, diffraction, polarization, and fluorescence, for example. Also using other sources, such as electron beams, to go beyond the phenomenological limits imposed by light sources. (2)

With the technological advances of electronic components, digital microscopy emerged, appropriating digital image sensors, which were originally simply added to the assembly of traditional microscopy. More recently, the significant increase in the number of digital cameras manufactured for use in cell phones has encouraged the improvement of digital image sensors and also contributed to lowering the prices of these devices. In this new scenario, lens-free microscopy was developed, enabling simpler, more robust, and portable instrumentation, to be used beyond laboratory benches. (3–5)

It is important to point out that, although many lens-free microscopes tend to have simplified instrumentation, which in fact do not use optical lenses throughout their assembly, this terminology is more broadly used. This is because, in the literature, the definition of a lens-free microscope is understood as equipment that provides amplified optical images, without the need for a lens between the sample and the sensor. Then, this interpretation also covers more complex systems, which may or may not have one or more

lenses in their setups. (6, 7) Among the lens-free microscopy of more simplified assembly, there is still a variety of configurations. To optically encode the scene information the system can generate a hologram provided by the light interaction with object elements from the scene or a pattern produced by an additional optical element, thus they can be illumination-modulated or mask-modulated, respectively. (8)

Considering only the illumination-modulated systems, a possible classification can be made based on the coherence degree of the lighting used. (9) This is because this characteristic of light defines the interference capacity between the light waves coming from the same source. If the light is completely incoherent, i.e., without the ability to provide interference, the instrumentation is based on the acquisition of images that represent the sample's shadow patterns, making it possible to classify the images, although they do not represent precisely the visual characteristics of the sample. Alternatively, if the microscope system has coherent light, the pattern recorded on the sensor corresponds to the interference of transmitted light waves with and without deviation by the sample. In this case, the registered image can be processed to obtain more detailed characteristics of the sample, making it possible to acquire amplitude and phase images of the same observed area. The intermediate case is characterized by the use of partially coherent light sources, which have a degree of coherence just enough to generate an interference pattern in which the sample information is recorded and, thus, reduce the influence of unwanted artifacts. The interference pattern of light that encodes the information of an object or a scene is called a hologram. For this reason, the lens-free imaging systems that use this encoded amplified information are called lens-free holographic microscopes.

Therefore, coherence is an important factor for assembling a digital holographic lens-free microscope, because the holograms, on which such instrumentation depends, are the recordings of interference patterns of the light that passes through the sample. Thus, the quality and resolution of images from this instrumentation are influenced by the coherence degree of the lighting used. And this, in turn, does not depend only on the chosen light source, but also on the components capable of filtering the light that comes from it, temporally and spatially. (4, 10)

In addition to the hardware of a lens-free digital holographic microscope, digital image processing also is necessary to perform images on this equipment. In this technique, digital image processing is used to decode holograms, which are recorded interference patterns. This processing, carried out after the acquisition of images, includes steps of image focusing and phase recovery to better characterize the light transmitted by the sample, at the instant of recording the hologram.

The potential of this instrumentation has been explored in several applications, presenting it as a very effective and versatile tool, being able to act from the monitoring of cells within controlled environments, such as cell incubators, (11) even in field research

for evaluation of the air pollution,(12) for example.

In order to be widely used and available in low-resource centers, these microscopes should be easily reproducible and rely on open-source image reconstruction codes. Although there is a great variety of lens-free holographic microscopes, and some advances to popularize this instrumentation, (13) there is still a lack of open-source codes for hologram reconstructions.

In this thesis, we present the development of two versions of lens-free holographic microscopes, based on the initial version of a lens-free microscope previously developed. (14) A different phase recovery method was applied for each one of these imaging systems. In addition to consolidating the domain of physics and digital processing involved in the assembly and operation of lens-free holographic optical microscopes, the objective of this study was to develop portable and easy-to-use platforms, based on the principle of lens-free holographic microscopy (LHM) with open-source codes, for imaging and monitoring biological samples.

2 OBJECTIVES

This study aims to develop portable and automated microscope instrumentation using a lens-free holographic microscopy technique based on open-source codes, besides exploring its applications. In order to promote comparison between some existing methods, this study proposes to develop two versions of the lens-free holographic microscope, using, in each one, a different phase reconstruction method: multiheight and multispectral. These two approaches have their own singularities both in hardware and software.

2.1 Specific objectives

- Hardware development of two lens-free holographic microscopes based on different phase reconstruction methods: multiheight and multispectral;
- Software development to image processing for each phase reconstruction method, based on open-source codes;
- Software development to control the microscope devices, based on open-source codes;
- Explore applications for the developed instrumentation.

3 LITERATURE REVIEW

Holography has its origin in 1948 when Dannis Gabor published his famous paper "A new microscopy principle" with the idea of a new technique to increase the resolution of electron microscopy. The principle was first tested in an optical model, using monochromatic light. The configuration was arranged in order to have the object positioned after the focal point of the light. Then, a photograph was recorded with the interference of the light from the source (reference wave) with the coherent part of the light emitted by the object (object wave). This photograph (now called hologram) was then illuminated with an optical imitation of the reference wave to allow one to see behind it the original object as if it were in place. (15)

The interference pattern was called "hologram", because "holos" in Greek means "whole" and the pattern contains the whole information, with amplitude and phase information of the light. (16) Because the hologram cannot distinguish positive and negative phase shifts with respect to the reference wave, a second image of the original object is found in a central-symmetrical position with respect to the focal point. This second image is called the "twin image" and is an artifact inherent to the holography. (15)

The original setup proposed by Gabor was called "in-line" because it uses the light aligned to the object and the photography plate. In this case, the reconstructed image is superimposed by the bright reconstruction wave and the twin image. (16) To overcome the limitations caused by this configuration, Leith and Upatnieks introduced an "off-axis" configuration, which provides the spatial separation of the two images and the reconstruction wave. (17)

Technological progress made it possible to perform the recording and reconstructing process computationally. Therefore, in digital holography, the photography plate was replaced by digital image sensors, and the physical reconstruction of the image was changed by numerical reconstruction. (17) Regarding recording digital holograms, photorefractive materials can also be used together with digital image sensors to record digital holograms with similar results. (18)

Digital holographic microscopy can provide 3D phase images from the observed objects by label-free and non-destructive measurements. In Brazil, this technique has already been applied to the analysis of microstructure surfaces and biological samples. (19, 20)

In its simplest configuration, digital holography has the same original assembly as the holography technique, called *in-line*, in which the components are all aligned. (21, 22)

Despite the existence of a plurality of holographic optical microscopes that use

digital image sensors to capture holograms to be decoded through numerical reconstructions, (21) it was in 2010 that this instrumentation became even simpler and designed to have a lower cost. (23)

This novel modality of digital holography has expanded the potential for applying the holograms, enabling the development and study of low-cost, lightweight, and portable assemblies for use in fluid analysis for the diagnosis of meningitis, (24) detection of parasites in the water, (25) cell motility analysis, (26) pathological histology (27) among other applications. (28)

Although the in-line configuration proved to be the least complex one to obtain holograms, other researchers have shown that it is possible to build portable and cost-effective instrumentation that makes use of reflected light to record the interference patterns (29,30) or even assemblies that allow transmission and reflection images, which has an in-line and off-axis configuration, respectively. (31)

Those assemblies that consider the light reflected by the sample are more restrictive regarding the size of the recorded image, while the transmission configuration may have greater freedom to determine the field of view (FOV), which is directly correlated with the distance between the sample and the sensor. Thus, the farther the sample is from the plane in which its image will be captured, the smaller the area observed and the greater the magnification of the scene. Taking into account this degree of freedom given by the distance between the sample and the image sensor, Tobon-Maya and his collaborators recently built a microscope with a variable field of view, maintaining the principles of expanding its accessibility. (13)

Considering one of the great benefits of these lens-free systems, which is the fact that their resolution and field of view are decoupled from each other, we built our instrumentation in order to take advantage of the largest possible area of our holograms, thus defining the sample position very close to the sensor. In this configuration, we have a practically unitary magnification and, therefore, a fully digital magnification.

In addition to choosing the size of the resulting images, the assembly of these systems is also defined by the methods used in digital processing. Therefore, the process of numerical light propagation and phase reconstruction, as well as the possible additional processing to increase resolution, for example, are dependent on the way in which the interference patterns are captured. Consequently, microscopes capable of generating a three-dimensional image with higher quality (32) or that are capable of obtaining a resolution below the pixel size, even with magnification equal to one, (33–36) have their particularities more evident.

Among the variety of algorithms used for processing lens-free microscope images, there is a growing use of machine learning for application in specific imaging conditions,

being used, for example, to count (37) and organize (38) cells, coloring images of histological slides to resemble those made in a brightfield microscope, with stained samples, (39) and even reconstructing phase values from the intensity information recorded in the holograms. (40, 41)

Methods of phase recovery based on traditional digital image processing use to be iterative and can be mask-based or mask-less. The first method needs no more than one hologram but requires previous information on the sample to construct a mask that determines where are the objects in the FOV.(42) The mask-less iterative phase recovery methods use a sequence of holograms of the same scene, recorded with different effective optical path-length. Therefore, the holograms can be recorded by varying the refractive index between the sample and the image sensor, the distance between these elements, the wavelength of the incident light, or even combining all these variable parameters. (43)

Considering mask-less iterative phase recovery methods, we highlight two: multiheight (44) and multispectral, (45) which are commonly used separately but can also be complementary, as shown in this assembly made by Bian and collaborators. (46)

Multiheight method uses a sequence of holograms recorded with different distances between the sample and the digital image sensor. Then, to have these images it is necessary to move the image sensor or the sample with some external device or add slides, for example, between them. Multispectral method commonly uses more than two different spectral ranges to illuminate the sample and produce the holograms. Both methods have been presented in literature and are implemented, separately, on the lens-free holographic microscopes developed in this study. Therefore, they are mentioned in more detail in Chapter 4.

Regardless of the method used to perform phase recovery, the tool used to make it work is of great importance. Many LHM systems rely on private or commercial codes. (44, 45) Recently, researchers have made more accessible versions of code available, such as an ImageJ plug-in made by Tobon-Maya and collaborators. (13) In this context, the present study intends to offer open codes to be easily accessed and complemented by any user. Thus, we decided to write these codes in Python, which is open-source and has been presented as a very versatile, friendly, and accessible computational language.

4 LHM BASE ON MULTIHEIGHT AND MULTISPECTRAL PHASE RECOVERY METHODS

4.1 Introduction

Lens-free holographic microscopy is a technique with great potential to offer simple and compact microscopy equipment. Besides the advantage of enabling a wide field of view, the LHM system also offers phase and amplitude information of the same scene with the use of phase recovery algorithms.(5, 41)

There is a vast variable of LHM configuration and methods applied for hologram reconstruction. These systems can be very compact and also have their hardware configuration projected to optimize their operation for the analysis of a certain type of sample.(12,47) Furthermore, important parameters such as spatial resolution can be improved with more complex hardware and software, designed to provide more information in the holograms used as input for digital image processing. (47–51) Considering software optimization, machine learning algorithms can also be applied to execute specific tasks very well. (52, 53)

In this study, we aim to develop portable and easy-to-use LHM systems able to reconstruct holograms from different samples with a wide field of view and a micrometer resolution.

In order to compare the performance of simple systems, this chapter will present the construction of two lens-free holographic microscopes and their reconstructed holograms. Both systems have their own hardware and software. However, some differences are considered in their settings to have the best and simplest version for each one.

4.2 Material and Methods

4.2.1 Assembly

The most simple and size-reduced versions of LHM systems are based on the in-line configuration. In this mode, all the components devices are vertically aligned. Thus, the holograms result from the light transmission, hence, they are the interference pattern of the light that passes through the sample without interaction and the light that interacted with it by diffraction, for example. Hence, the intensity $I(\vec{r})$ of the light that reaches the digital sensor can be understood as the superposition of two light waves, described by their electrical fields: the reference wave $E_{\text{ref}}(\vec{r})$ (which has no sample information) and the light scattered by the sample $E_{\text{scat}}(\vec{r})$:

$$I(\vec{r}) = |E_{\text{ref}}(\vec{r}) + E_{\text{scat}}(\vec{r})|^2 = |E_{\text{ref}}|^2 + |E_{\text{scat}}|^2 + E_{\text{ref}}(\vec{r}) \cdot {E_{\text{scat}}}^*(\vec{r}) + {E_{\text{ref}}}^*(\vec{r}) \cdot E_{\text{scat}}(\vec{r}) \quad (4.1)$$

where ${E^*}_{\text{ref}}(\vec{r})$ e ${E_{\text{scat}}}^*(\vec{r})$ are the complex conjugates of the reference wave $E_{\text{ref}}(\vec{r})$ and the wave scattered by the sample $E_{\text{scat}}(\vec{r})$, respectively.

The first term on the right-hand side of the equation 4.1 ($|E_{\text{ref}}|^2$) is the background intensity, related to the reference wave. The second term ($|E_{\text{scat}}|^2$) is the intensity of the scattered wave. The last two terms ($E_{\text{ref}}(\vec{r}) \cdot {E_{\text{scat}}}^*(\vec{r})$ and ${E_{\text{ref}}}^*(\vec{r}) \cdot E_{\text{scat}}(\vec{r})$) are the holographic terms that contain information about the sample images. In the condition that the sample has low scattering, $|E_{\text{scat}}|^2$ is negligible in comparison with the other terms, which is commonly assumed in holography.

A typical issue of holography is presented by the holographic terms: the duplication of the image with sample information. It occurs because the digital image sensor only records the intensity of the incident light, consequently causing the loss of phase information. These duplicated (real and virtual) images are called twin images because they have the same information but different plane positions. In the in-line configuration, the reconstruction process superposes both images and causes undesirable artifacts, which can be reduced or even suppressed by applying phase recovery methods. The distance between the twin images is $2 \times z_2$ because the positions are symmetric with respect to the plane of the hologram. Therefore, using the configuration to have the larger FOV (with a small z_2 distance), the twin image is close to the original image. This makes the application of phase recovery even more important in this case.

Considering the in-line configuration to record holograms, the sample-sensor distance controls the amplification of the projected light onto the sensor. This parameter, together with the active area of the image sensor will define the field of view (FOV) of the LHM system. The closer the sample is to the sensor, the larger the FOV. Therefore, in all the presented microscopes, the sample is positioned as close as possible to the digital image sensor, which will capture the interference pattern of the light transmitted and diffracted by the sample. Such a setup, as shown in Figure 1, with the distance between the light system and sample much greater than the distance between sample and image sensor ($z_1 >> z_2$), resulting in a hologram with an almost unitary magnification. In this condition, FOV has approximately the same dimensions as the active area of the digital image sensor. Figure 2 shows the visual comparison of the FOV of the presented microscopes and images of the same sample obtained with objective lenses of different magnifications.

The in-line configuration was applied in all the lens-free holographic microscopes developed in this study. This model is also used in the previous version of the LHM system developed in the early stages of this study.(14) This previous assembly was bench

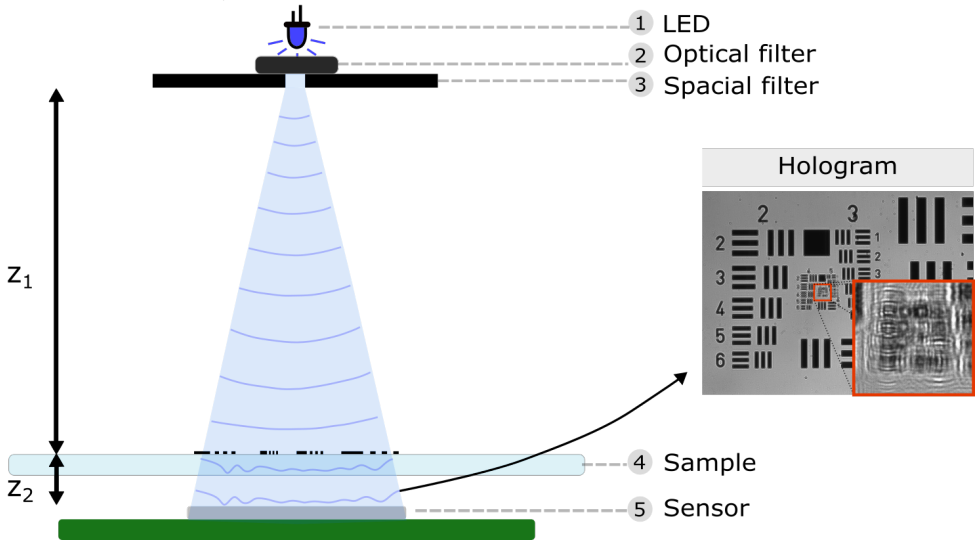

Figure 1 – Representation of a lens-free holographic microscope in the in-line configuration with its component devices and typical hologram using a USAF-1951 resolution test target as a sample.

Source: By the author.

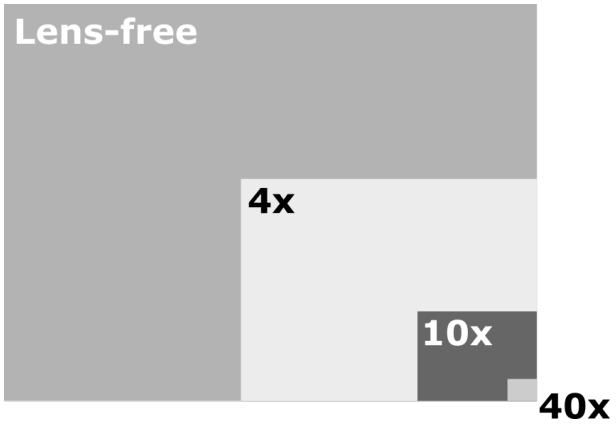

Figure 2 – Comparison between the field of view (FOV) area of the lens-free holographic microscopes developed in this study with the FOV of the traditional bright field microscopes with an objective lens of 4×, 10×, and 40×.

Source: By the author.

equipment based on the multiheight phase recovery method. It required manual operation and had only the necessary digital image processing performed by Matlab (54) algorithms.

In addition to the multiheight method, another phase recovery method was implemented: the multispectral method. As they have different imaging acquisition processes, in this study, a microscope system was developed for each method.

Lens-free holographic microscopes presented here were projected to be portable, compact, cost-effective, and easy-to-use. Thus, hardware and software modifications were implemented to optimize the image results, besides promoting system automation.

4.2.2 Numerical propagation of light

All digital image processing described for holograms in the present study is based on the numerical propagation of light. Since $z_1 >> z_2$, light that achieves the digital imaging sensor can be considered a plane wave. Thus, the Angular Spectrum Method (ASM) is used to calculate the light propagation.

ASM comes from the scalar diffraction theory and interprets the electric field using plane wave decomposition, by Fourier Transform. Thus, the amplitude of the electrical field in a spatial point is given by the sum of the contributions of these plane waves. (55, 56)

A decomposition of an electric field $E(x_0, y_0)$, which is on a plane $z = 0$, is made by the following equation, where $A_0(k_x, k_y)$ is known as Angular Spectrum:

$$A_0(k_x, k_y) = \mathcal{F}[E(x_0, y_0)] \tag{4.2}$$

$$A_0(k_x, k_y) = \frac{1}{2\pi} \iint_{\Sigma_0} E(x_0, y_0) e^{-i(k_x x_0 + k_y y_0)} dx dy \tag{4.3}$$

where k is the wave vector, with magnitude $\frac{2\pi}{\lambda}$, in which λ is the wavelength of the light, k_x and k_y are the components of the wave vector k, and $\mathcal{F}$ is the Fourier Transform and Σ_0 the initial plane.

From the decomposition of the initial field $E(x_0, y_0)$, it is possible to calculate the same electrical field in another plane parallel to the initial one, $E(x, y, z)$:

$$E(x, y, z) = \mathcal{F}^{-1}[A_0(k_x, k_y) e^{i\sqrt{k^2 - k_x^2 - k_y^2} z}] \tag{4.4}$$

$$E(x, y, z) = \frac{1}{2\pi} \iint_{\Sigma_0} A_0(k_x, k_y) e^{i(k_x x_0 + k_y y_0) + \sqrt{k^2 - k_x^2 - k_y^2} z} dk_x dk_y \tag{4.5}$$

where $\mathcal{F}^{-1}$ is the inverse of Fourier Transform, and $k^2 \geq k_x^2 + k_y^2$.

The description made by this method is equivalent to the Rayleigh-Sommerfeld solution and does not depend on a minimum propagation distance. This approach is commonly applied to reverse light propagation in microscopes similar to those we are presenting. (44, 46)

4.2.3 Autofocus

A fundamental step for a good performance of digital processing for holograms is the definition of the optical path between the sample and the digital image sensor. In a homogeneous medium, as this is considered, this term is defined by multiplying the index of refraction of the medium (n) by the distance traveled by the light (z_2). This term is essential to describe the numerical propagation of light, and consequently, to predict the image information in the sample plane, from recorded holograms. Despite the absence of lenses in these microscopes, this stage of digital processing is called focusing. It is because this step consists of the numerical propagation of the original hologram (an image with pronounced artifacts from diffraction effects) to a plane in which the sample image is given with a higher definition.

The sample-sensor distance and the refractive index of the medium are known experimentally with some approximation. However, the value of the optical path must be defined with accuracy to ensure a better resolution for the reconstructed holograms. In the digital processing, we fixed the refractive index value and calculated the focal length from a parameter that presents a maximum value for the searched distance.

Researchers evaluated the performance of several methods for the automated calculation of this focusing distance and demonstrated superiority in the results obtained with the use of the Tamura Coefficient (TC).(57) TC was applied to perform the autofocusing of the holograms from LHM, evaluating the complex information of the images calculated from holograms propagation, as described by equation 4.6:

$$TC(|\nabla E|) = \sqrt{\frac{\sigma(|\nabla E|)}{< |\nabla E| >}} \tag{4.6}$$

where E is the complex electric field matrix and σ is the standard deviation, in this case, calculated for the values of the matrix that represents the modulus of the E gradient ($|\nabla E|$) and $< |\nabla E| >$ represents the mean of these values.

To define the best value to describe the z_2 distance, the original hologram is cropped to restrict the area of the image to be analyzed. Each of these cropped images is propagated over a predefined range of distances. The TC value is then calculated with the gradient modulus of each propagated image. The highest value of this TC parameter will be related to the image with the highest definition and, thus, the distance at which it

was propagated will be defined as the focus. Figure 3 summarizes the steps of our digital processing for the automatic focusing of hologram images.

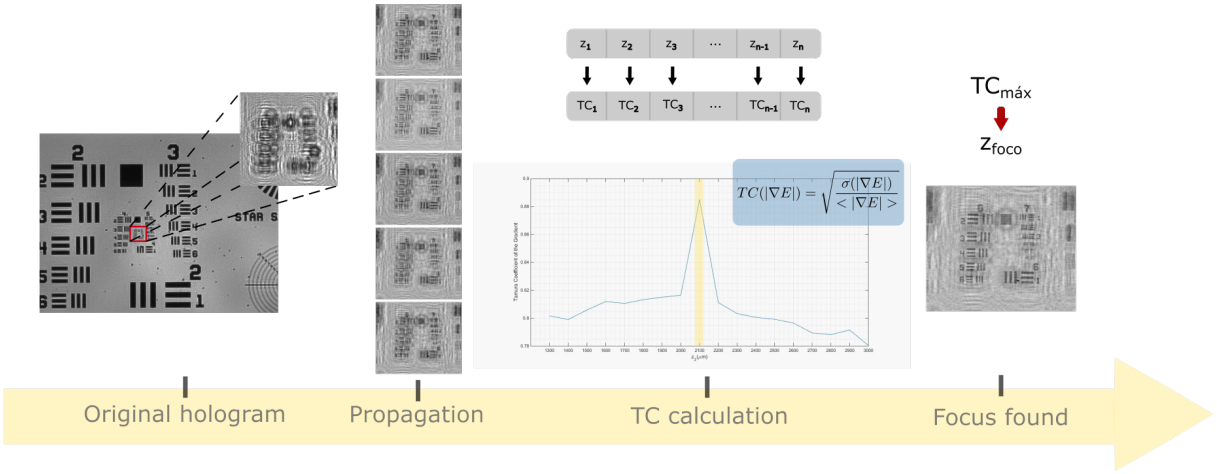

Figure 3 – Steps for the autofocus imaging processing applied to the holograms, using Tamura Coefficient (TC) calculation. The hologram shown as an example was made using a USAF-1951 resolution test target as a sample.

Source: By the author.

4.2.4 Resolution

The components of the LHM system directly influence the spatial resolution of their images. Therefore, it is essential to understand the contributions of each factor to optimize the construction of the equipment with satisfactory spatial resolution in its holograms and, consequently, in the reconstructed images.

The main factors that influence the resolution of lens-free digital holographic microscopes are:(58)

- (1) sample-sensor distance (z_2);
- (2) temporal and spatial coherence of the light source (dependent on temporal width $\Delta\lambda$ and spatial filter diameter D, respectively);
- (3) finite pixel size;
- (4) finite size of the processed image.

The resolution of the reconstructed images depends on the pixel size of the images, which is the same for the holograms. Therefore, in all the experiments shown in this study, we used the same sensor with reduced pixel dimensions ($1.67\mu m \times 1.67\mu m$).

To ensure the proximity of the sample to the image sensor (shorten the z_2 distance), the optical components and their holders were removed from the original camera system used in all the presented lens-free microscopes (Arducam CMOS MT9J001 1/2.3-Inch 10MP Monochrome Camera Module; Arducam USB2 Camera Shield). Just the thin glass ($\sim 0.4mm$) was kept over the sensor to protect it.

The coherence degree of the light source is also an important factor for the image resolution and directly influences the presence of artifacts. Nonetheless, the high coherence factor can cause artifacts due to excessive interference. Considering this, there are LHM systems that implement the use of modified lasers in order to reduce their coherence.(59) Another alternative applies filters to improve the coherence factor of a LED, as applied here.(60, 61)

The spatial resolution can be calculated from the spatial and temporal coherent length, independently limiting the resolution of lens-free holographic imaging. (10) It is important to note that the calculation of the spatial resolution, as well as its experimental determination, must follow some criterion, as is the case of the Abbe criterion, used in equations presented by Ozcan and McLeod. (10) However, most studies do not clarify the criteria used for detecting spatial resolution experimentally, which makes it difficult to compare results.

For the comparison experiments between the implemented methods, we used equivalent factors but optimized to improve the resolution of our holograms in each case. Some tests were done in order to define the elements that provide the best resolution for the lens-free microscopy instrumentation. Thus, maintaining the LED light, the resolution was analyzed by changing the diameter of the spatial filter (D), the distance between light and sample (z_1), and the spectral width of the light source ($\Delta\lambda$) with or without the addition of an optical filter.

Therefore, considering the optimization and simplicity of both assemblies, light coherence was increased in a particular way in each case. Although we use lighting from LEDs with similar spectral widths ($\Delta\lambda \approx 20nm$, considering the full width at half maximum - FWHM), the final version will have the addition or not of an optical filter. Besides this, for both methods, we also used a circular aperture spatial filter with diameter $D = 50\mu m$ and we kept the same holders for the sample, as these directly affect the distance z_2. In addition, image processing was always done with the same dimensions as the initial holograms. Then, for the results presented here, these dimensions correspond to the entire field of view.

4.2.5 Sample

To define experimentally the spatial resolution for the reconstructed hologram, a positive USAF-1951 test target (R1L1S1P, Thorlabs, USA) was used. This target is stan-

dard to measure the spatial resolution for microscopy optical assemblies. It has spatial patterns of standardized size, which reduce according to the number of groups they belong to. Thus, the spatial resolution can be measured based on the size of the smallest distinguishable patterns in the image.

The resolution defined by the USAF-1951 test target is often expressed in terms of line pairs per millimeter (where a line pair is a sequence of one black and one white line), which is a spatial frequency. The inverse corresponds to the distance between two resolved lines. (62)

4.2.6 Multiheight

4.2.6.1 Hardware

The final version of the light system of the multiheight lens-free holographic microscope is composed of a single LED with emission centered at 455 nm and spectral width FWHM of 18 nm, which had this temporal coherence enhanced by the addition of an optical filter (with $2nm$ of spectral width FWHM) and its spatial coherence improved by the use of a pinhole (50 μm of diameter).

As mentioned before, this equipment uses in-line configuration to record their holograms. Thus, in the multiheight lens-free holographic microscope, all the components are vertically aligned in the following order, as shown in Figure 4:

- (1) LED (M455L3, Thorlabs, USA);
- (2) Optical filter (LL01-442-12.5, Semrock, USA);
- (3) Spatial filter (P50S, Thorlabs, USA);
- (4) Sample;
- (5) Digital image sensor (MT9J003, Aptina/Onsemi, USA).

The first version of the presented multiheight lens-free holographic microscope was bench equipment. In order to have a portable, compact, and easy-to-use system, some modifications were made to the hardware. Then, a 3D-printed structure was designed to accommodate all the components and ensure the z_1 distance is around $5cm$.

This 3D-printed structure has two parts: the base, inside which is the camera system and the LED electronic circuits; and the cover, which holds the lighting system and fits into the base to block the external light. As illustrated in Figure 4, the digital image sensor was exposed outside the top of the base, on which the sample will be placed.

To obtain each reconstructed image in the multiheight method, it is necessary to record a sequence of holograms of the same scene, at different sensor-sample distances.

These images are then used as input data for digital processing. These step distances have to be short enough to ensure that the magnitude of the recorded holograms remains close to one.

To promote a precise movement of the image sensor, a motorized actuator was used to move the platform that holds the camera system. $12mm$ screwdriver (Z812B, Thorlabs, USA), connected to a servo motor (K-Cube Brushed DC Servo Motor Controller, Thorlabs, USA) is responsible for changing the height of the sensor in relation to the sample, which in turn is held in the same position throughout the acquisition. Experimentally, it was defined the step distance (Δz) is $50\mu m$, and six is the total number of holograms for each sequence.

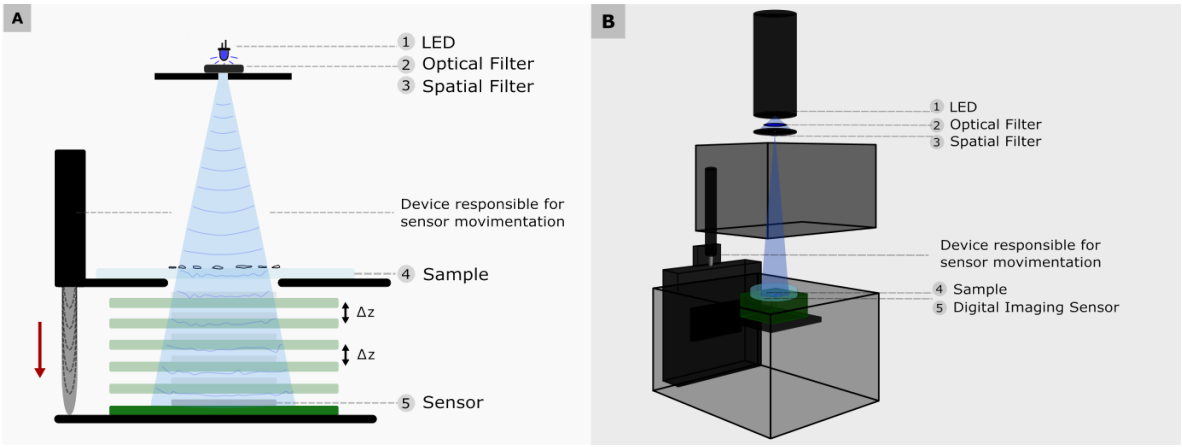

Figure 4 – Schematic representation of multiheight lens-free holographic microscope with its component devices in (a) 2D and (b) 3D.

Source: By the author.

4.2.6.2 Software for phase recovery

As the holograms were recorded at different heights (related to the sample position), they have slightly different optical paths. Therefore, this information is used to converge the phase values in the calculations.

Before performing this phase recovery process, the raw holograms are aligned in order to offer the best performance for the final reconstructed image. To reduce the computational cost, this imaging process is performed on a cropped region, with reduced dimensions compared to the original one. So, the alignment is applied in a selected area of the in-focus holograms, performing vertical and horizontal shifts, to adjust their position according to the first one. Then, the shifting parameters are applied to the original (and entirely) holograms that are used for further image processing.

The multiheight phase recovery has been used since the first version of our microscope. (14) It is based on the Gerchberg-Saxton algorithm, whose calculation predicts phase distribution from known intensity images in two parallel planes.(63) The amplitudes are proportional to the square roots of the measured intensities. The initial estimation for phase distribution is assumed to be an array of random numbers between π and $-\pi$. With this initial information, the phase values prediction is performed by numerical propagation of complex values from one imaging plane to another. After each propagation, the complex number that describes the recorded light is updated. Thus, new phase values are used together with the measured amplitude related to that image. Some iterations are performed to gradually converge the phase values to the correct ones.

The phase recovery method applied in this study is a modification of the Gerchberg-Saxton algorithm as presented by Greenbaum *et. al.* (48). Some variables (as Δz and the number of holograms) were tested and it was consolidated the use of this method using six original holograms and step distance (Δz) equal $50\mu m$. The hologram acquisition begins recording the first image closest to the sensor and the others with $50\mu m$ steps away from it.

As described by Greenbaum *et. al.*, (48) the initial phase values are assumed to be zero and are modified over iterations. For the presented algorithm, each iteration consists of applying numerical propagation from the first to the last recorded hologram and back-propagating the holograms from the last to the first. Ten iterations are enough to converge the phase values, resulting in a better-quality reconstructed image at the sample plane.

Figure 5 resume visually the image acquisition and processing performed on holograms in this multiheight method.

4.2.6.3 Software for device control

Originally, multiheight lens-free holographic microscope devices were controlled separately by programs provided by the respective manufacturers. This condition implies the use of different computer programs and the manual introduction of their parameters. Therefore, it made the equipment difficult to use, besides increasing the time of the hologram acquisition process.

The original programs for each one of the applied devices are presented below:

- Motorized actuator - Zadig η(available on $https://zadig.akeo.ie/$)
- Arducam camera - GUI for Windows (available on GitHub $https://github.com/ArduCAM/ArduCAM_USB_Camera_Shield/tree/master/Windows$)

In order to promote the easy use of the presented equipment, an algorithm was developed to integrate these device commands in a single computational program, using

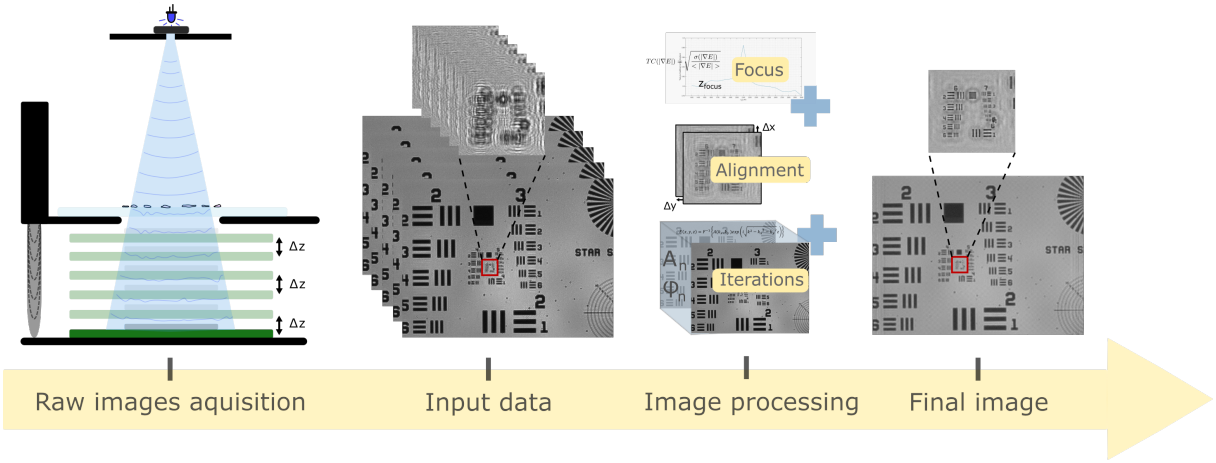

Figure 5 – Image acquisition and processing steps of the Multiheight method for LHM. The hologram shown as an example was made using a USAF-1951 resolution test target as a sample.

Source: By the author.

Python language. The documents by the manufacturers made it possible to access the computer commands related to each of these devices' control. To move the motorized actuator and access its displacement parameters, by Python algorithm, it was used the following libraries: *pyAPT*, *libftdi1*, and *pylibftdi*. Also, the use of Anaconda platform (Austin - Texas, USA) (64) was required. Functions provided by Arducam, using *ArducamSDK*, *json*, *Opencv*, and *json* libraries, were applied to control the camera system.

The *threading* library was used to make functions work in parallel, to properly control the camera system. Additionally, the *flag* concept was applied to activate and deactivate functions that control the devices. *Flag* is applied by a boolean variable that signalizes if part of the computational program should run or not, accordingly with this value (*True* or *False*).

Figure 6 represents this software developed in Python to control the actuator and camera devices.

The initial version of this software has a simple interface, which works on the Windows command prompt. A menu displays the control options for use. Each option is related to a letter and is activated by typing the letter and confirming using the *Enter* key, as shown in Figure 7.

Other versions were developed for this software control, in order to implement improvements. One of them was the addition of a new option, which allows the temporal monitoring of a sample by LHM holograms. It was done by automated control of a sequence of multiheight hologram acquisitions, with quantity and time intervals defined by the user.

Figure 6 – Schematic representation of hardware and software related to the hologram acquisition on multiheight lens-free holographic microscope. The red devices on the microscope represent the camera and translator, which are connected to the computer to automate the acquisition process.

Source: By the author.

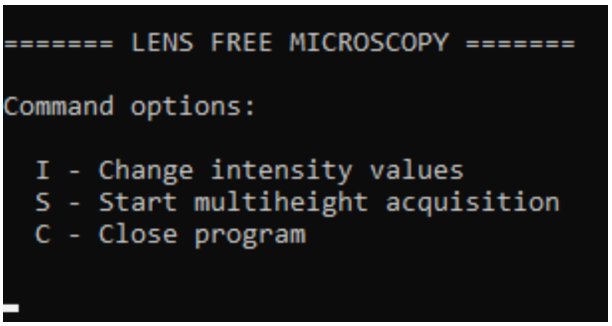

Figure 7 – Menu options of the software developed for device control of the multiheight LHM system.

Source: By the author.

A graphical user interface (GUI) was implemented on the final version of this software, using *TKinter* library. Figure 8 presents the operating steps in this algorithm.

4.2.7 Multispectral

4.2.7.1 Hardware

The first version of the multispectral LHM system developed in this study used the light from a LED lamp (7,5 W, A60 RGB, Osram, Germany) with light emission centered at 634nm, 518nm, and 462nm and spectral width of 21nm, 33nm, and 23nm for red, green and blue LED, respectively. This initial configuration was used for tests, presented in Chapter 5, performed to compare the applications from the multispectral and multiheight systems developed in this study.

Regarding the issues of this first version, a new illumination system was developed in order to improve the physical connections of the hardware, besides contributing to the

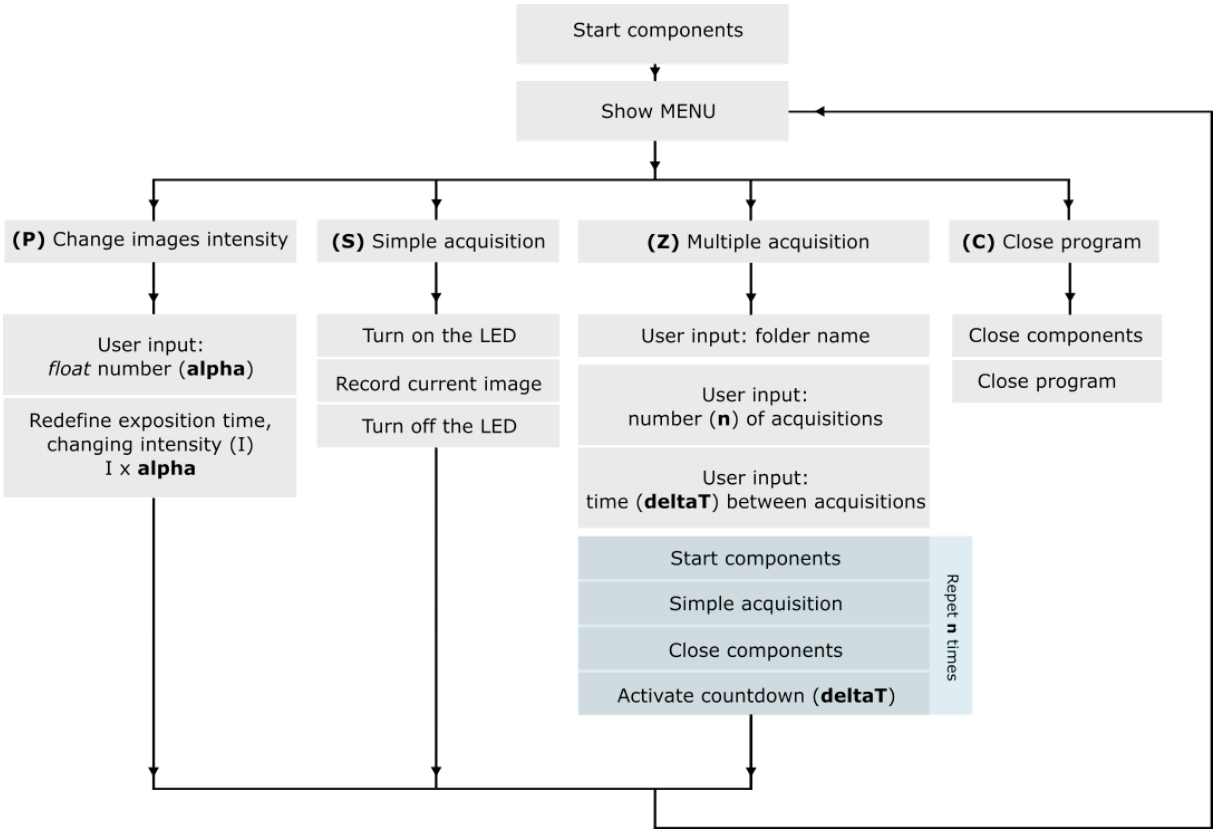

Figure 8 – Schematic diagram of the operating steps of the control software developed for the multiheight LHM equipment.

Source: By the author.

automation of this instrumentation. Therefore, high-power RGB LEDs were implemented in a new illumination system. The illumination is centered at $632nm$, $518nm$, and $453nm$ and has a spectral width FWHM of $17nm$, $34nm$, and $21nm$, respectively for red, blue, and green LED.

An electronic circuit was constructed for each one of the RGB LEDs and connected to a microcontroller (ATmega32u4, SparkFun, USA). The use of a microcontroller makes it possible to control each LED (turn it on or off), independently, using computational commands. A USB cable makes the connection between the microcontroller and the computer. An aluminum heatsink was projected using Inventor software (Autodesk, San Rafael-CA, USA) to be in contact with the LEDs in order to ensure the proper functioning of the illumination system.

Another important improvement performed on this microscope version was the addition of an electronic relay to physically control the connections between the camera and the computer. Without this electronic device, the camera connection, provided by a USB cable, maintains the camera's power supply, even if there is no connection with the

computer software. This implies an increase in temperature in the region of the active area of the digital image sensor. Therefore, this limitation restricts the use of the system on samples that are not sensitive to temperature variations of approximately 15^oC. With the addition of the electronic relay, the temperature rise could be minimized by turning the camera on only when it is actually being used. This provides temperature regulation and helps to expand the possibilities of using this microscope.

As the multiheight LHM equipment, the multispectral lens-free holographic microscope developed in this study has all their components vertically aligned in the following order, as shown in Figure 9:

- (1) RGB LEDs (homemade system with high power LEDs with 3V);
- (2) light diffuser;
- (3) Spatial filter (P50S, Thorlabs, USA);
- (4) Sample;
- (5) Digital image sensor (MT9J003, Aptina/Onsemi, USA).

The high-power LEDs are required because part of the emission intensity is blocked by the spatial filter.

The light diffuser is very important here because the LEDs are positioned around the central point, which is aligned with the spatial filter. Thus, in absence of the light diffuser, the light from any RGB LED does not reach the digital sensor. It is also important that this component has a high transmittance to not require a further increase in the light intensity of the LEDs or an increase in the camera exposure time.

In this microscope system, to keep it simple, there are no optical filters. The addition of optical filters would not be as simple as in the case of the multiheight method. In this case, there would have a specific optical filter for each spectral region, besides a mechanical mechanism should be used to perform a rotation in order to ensure the alignment of each filter at a time.

To provide one reconstructed hologram, this multispectral LHM system record 3 holograms of the same scene, using different spectral range illumination for each one. Therefore, in this case, the acquisition imaging process results in 3 grayscale holograms obtained with red, green, and blue light from RGB LEDs.

4.2.7.2 Software for phase recovery

The multispectral method has, as input for its processing, holograms acquired by light sources with emission centered at different spectral ranges. Thus, to acquire these

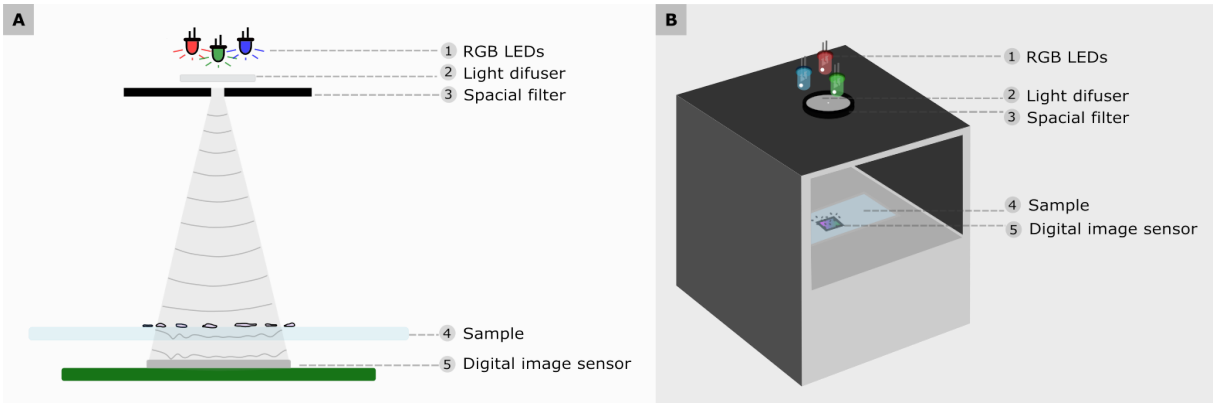

Figure 9 – Schematic representation of multispectral lens-free holographic microscope with its component devices (a) in 2D and (b) 3D.

Source: By the author.

images, we used a homemade RGB LED system with wavelengths centered at $462nm$, $518nm$, and $634nm$. As in the other method, the original holograms are aligned before being used as input for this image process.

Also for the multispectral method, the iterations are based on numerical propagation between the acquisition plane and the sample plane, with each of these iterations starting with updating the amplitude and phase values, presented by equations 4.7 and 4.8, according to the procedure described by Allier et al. (45) Every time the image returns to the initial plane, one iteration is counted. After 10 iterations there is a notable improvement in the quality of the images. For updates of the amplitude (A) and phase (ϕ) values of each of the wavelengths used (λ_j), we then have:

$$\phi_{new}^{\lambda_j} = \frac{1}{3\lambda_j} \sum_{i=1}^{3} \phi^{\lambda_i} \lambda_i \tag{4.7}$$

$$A_{new}^{\lambda_j} = \frac{1}{3} \sum_{i=1}^{3} A^{\lambda_i} \tag{4.8}$$

where the *new* index represents the updated value of each variable and the index λ_i refers to the parameters obtained with each of the used wavelengths, being the quantity i determined by the number of different wavelengths used. We use illumination at spectral well-separated wavelengths to be able to suppress the extra twin image contribution from updates with equations 4.7 and 4.8 since the dependence with respect to the wavelength is larger for the twin image than for the complex image of the object. (45)

Figure 10 resume visually the image acquisition and processing performed on holograms in this multispectral method.

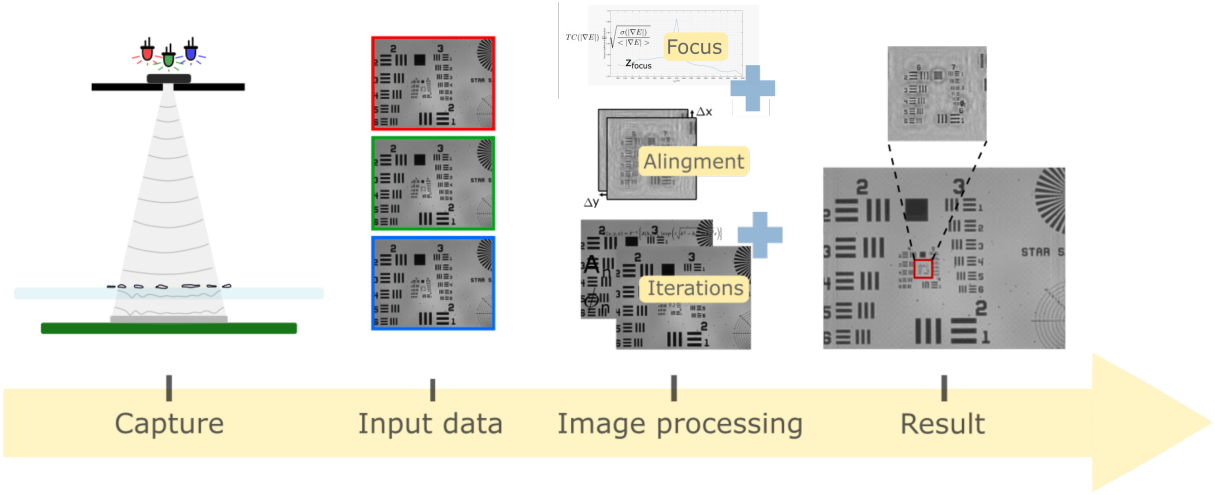

Figure 10 – Image acquisition and processing steps of the Multispectral method for LHM.

Source: By the author.

4.2.7.3 Software for device control

The automation of the multispectral LHM equipment, represented in Figure 11, was performed by software developed in Python to control the camera and RGB LEDs. To perform the camera control, the Python algorithm uses the same libraries presented before (*ArducamSDK*, *json*, and *Opencv*). And the control of the RGB LEDs depends on the *serial* library.

This software has a simple interface that works on the command prompt of Windows, displaying a menu with presenting the options for the user, as shown in Figure 12. Each option was activated by typing the corresponding letter and confirming using the *Enter* key.

The first option (I) allows the user to control the light intensity of the hologram, by changing a variable related to the exposure time of the camera. As Figure 13 shows, the sensitivity of the monochrome image sensor depends on the incident wavelength. Thus, it was previously defined the initial exposure time values for each illumination color as a vector $[exp_1, exp_2, exp_3]$. The relation between exp_1, exp_2, and exp_3 also depend on the absorption spectra of the sample, in order that it can be adjusted for each situation. Then, the *intensity* variable is defined as the multiplication factor applied on this vector: $new\_exposition\_values = intensity \times [exp_1, exp_2, exp_3]$.

The preview (P) shows the hologram using the blue LED to illuminate the sample. The holograms acquisition was performed by (S), which records 3 holograms: one red, one green, and one blue. (C) option is used to close the program. Figure 14 shows the software steps for each of these command options offered to the user.

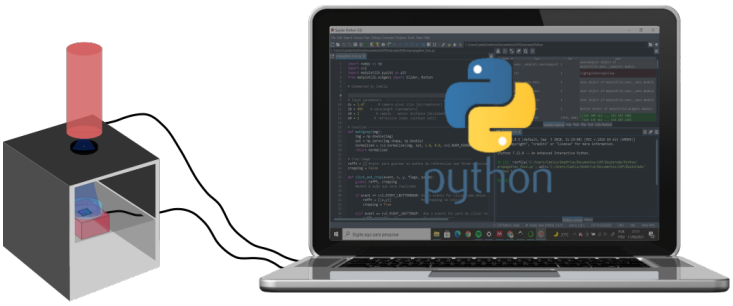

Figure 11 – Schematic representation of hardware and software related to the hologram acquisition on the multispectral lens-free holographic microscope. The red devices on the microscope represent the camera and LEDs, which are connected to the computer to automate the acquisition process.

Source: By the author.

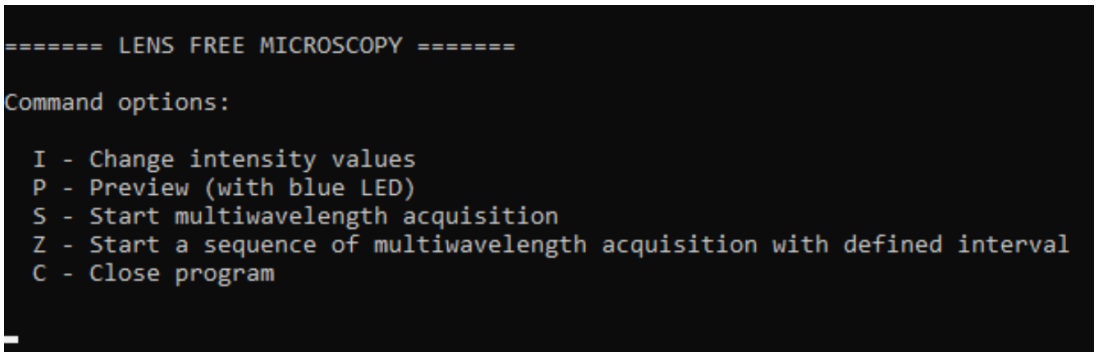

Figure 12 – Menu options of the software developed for device control of the multispectral LHM system.

Source: By the author.

4.3 Results and Discussion

4.3.1 Multiheight

Figure 15 presents an example of hologram reconstruction using the multiheight method. As can be seen in the amplified region, diffraction is evidenced in the smallest spatial patterns of the test target. The resolution limit of the presented imaging techniques relies on (or nearby) the spatial frequency of the group 7 elements of the USAF-1951 test target. Thus, these elements have great relevance for comparing the resolution of the images presented in this chapter.

Initial tests were performed to confirm the elements to be used in the LHM assembly and also to provide a visual difference caused by some of the parameters that influence the spatial resolution of the LHM systems. For each case, six holograms were registered using the blue LED (with emission centered at 455 nm) and were submitted to

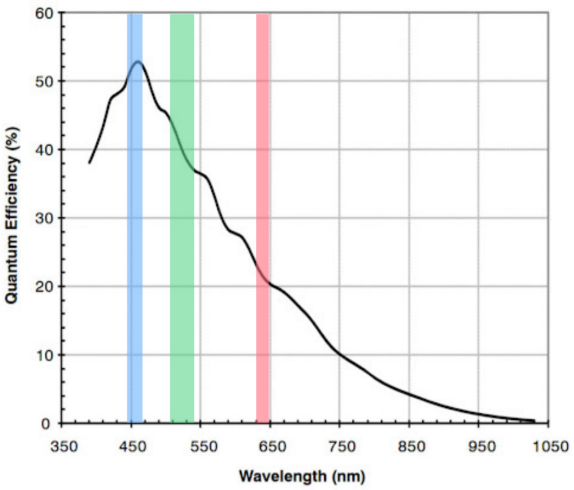

Figure 13 – Graph of the quantum efficiency of MT9J001 digital image sensor as a function of the wavelength. The blue, green, and red regions in the graph represent the emission range of the RGB LEDs applied in the multispectral system.

Source: Modified from MT9J001 CMOS datasheet.

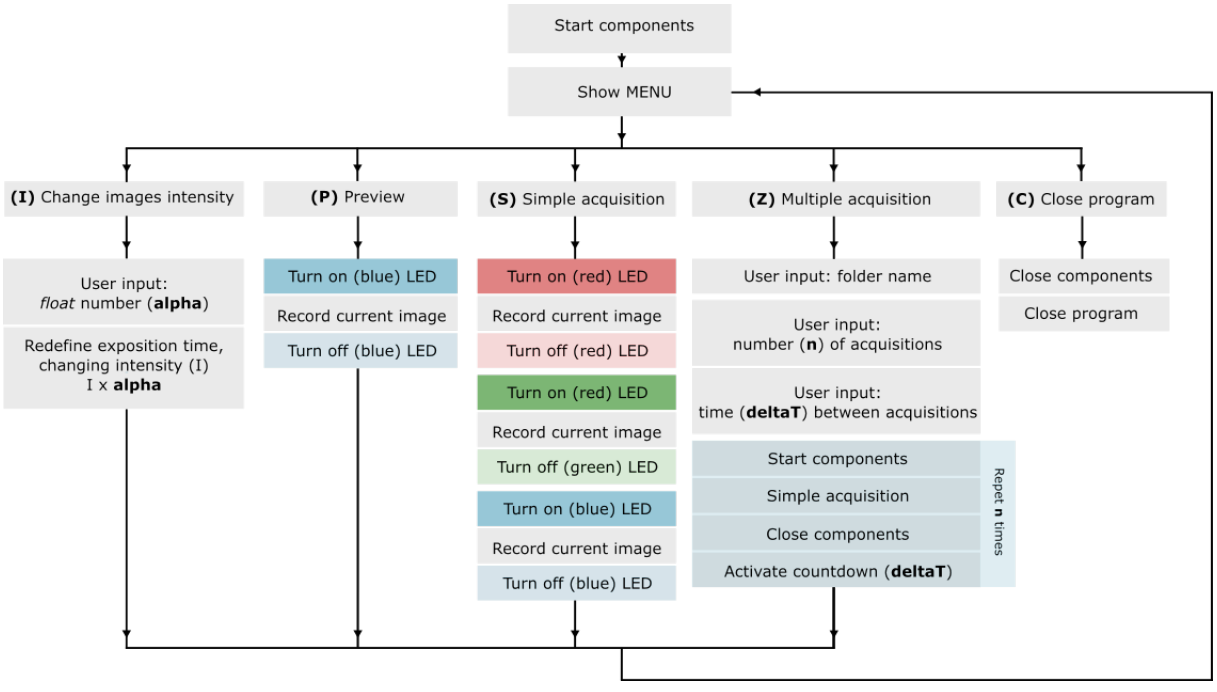

Figure 14 – Schematic diagram of the operating steps of the control software developed for the multispectral LHM equipment.

Source: By the author.

the multiheight phase recovery method.

Figure 16 shows the amplitude images resulting from the holograms recorded with lower spatial coherence, using a pinhole with a diameter (D) of 150μm. Figure 17 presents the amplitude images recorded with higher temporal coherence of the light, using a pinhole with $D = 50\mu m$. To promote the visual analysis, both Figures (16 and 17) show only the region of the images corresponding to the spatial patterns of the elements belonging to groups 6 and 7 of the resolution test target USAF-1951. Despite being a small region of the entire image (as shown in Figure 15), it represents the resolution of the entire imaged area. Also, the intensity profile is presented for the smallest element (element 6 of group 7) for all conditions. As the reconstructed holograms are 8-bit images, the gray values, related to the intensity of each image, rely on the range from 0 to 255.

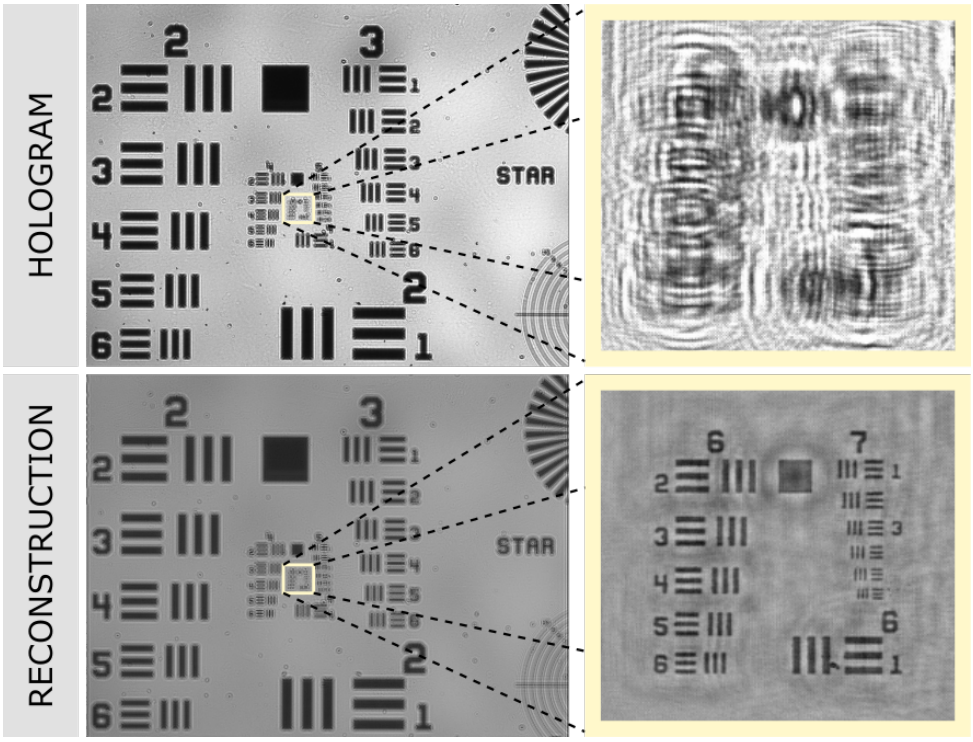

Figure 15 – Example of a hologram of a USAF-1951 resolution test target from a LHM system and its image reconstruction by multiheight method. In this case, the amplitude image (represented as reconstruction) results from the image processing using 6 holograms, recorded with these following parameters: $\lambda = 455nm$, $\Delta\lambda$(FWHM)$= 2nm$, $D = 50\mu m$, $z_1 = 5cm$, $z_2 \approx 2mm$.

Source: By the author.

The images presented in Figures 16 and 17 show the expected result: spatial resolution increases with increasing temporal and spatial coherence, given by the reduction of $\Delta\lambda$ and D, respectively. They also reinforce that the relation $z_1 << z_2$ is satisfied for $z_2 \approx 2mm$ and $z_1 \approx 5cm$. Therefore, regarding the compactness of the imaging system, the minimum z_1 distance tested was defined to characterize our assembly.

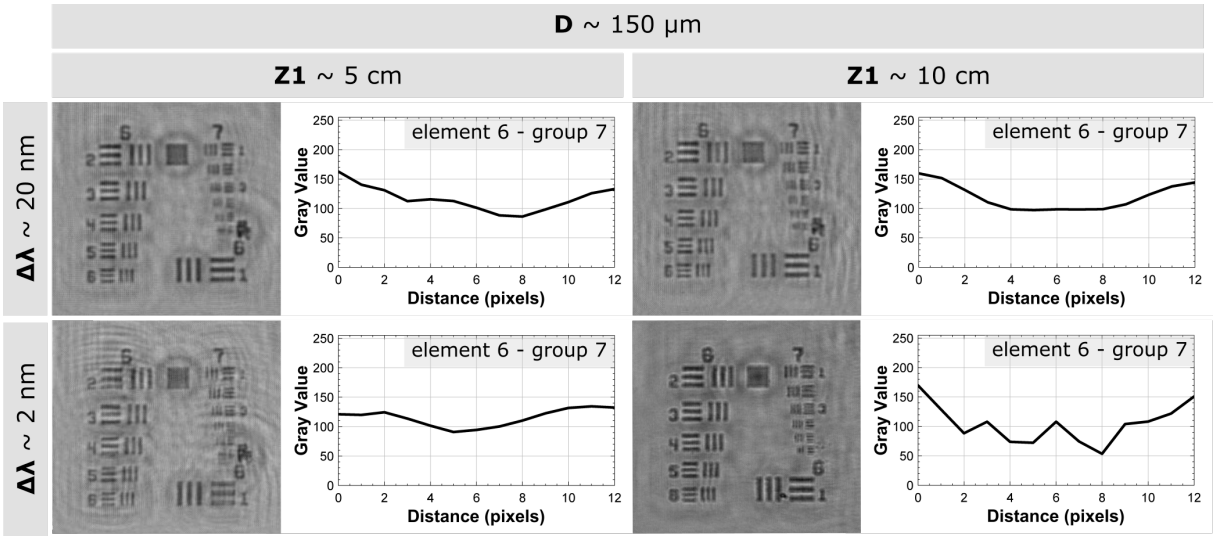

Figure 16 – Cropped amplitude images resulting from multiheight lens-free holographic microscope using LED with emission centered at 455 nm, pinhole with a diameter (D) of 150μm and different light-sample distances (z_1) and spectral width ($\Delta\lambda, FWHM$) as shown. The intensity profile of the horizontal smallest elements is shown next to each one of the images.

Source: By the author.

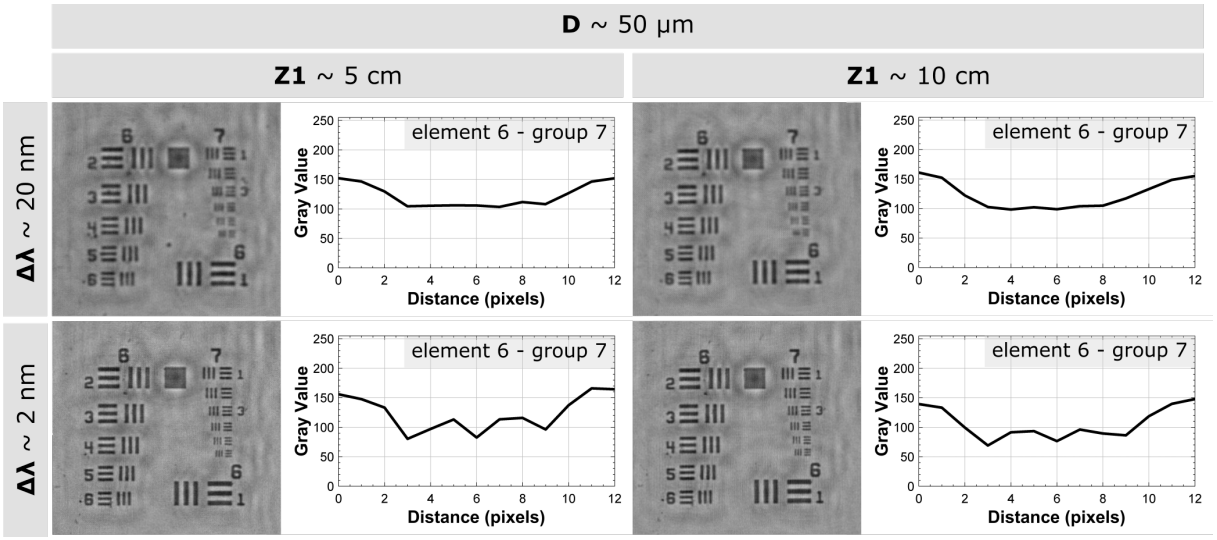

Figure 17 – Cropped amplitude images resulting from multiheight lens-free holographic microscope using LED with emission centered at 455 nm, pinhole with a diameter (D) of 50μm and different light-sample distances (z_1) and spectral width ($\Delta\lambda, FWHM$) as shown. The intensity profile of the horizontal smallest elements is shown next to each one of the images.

Source: By the author.

Considering the chosen parameters, a theoretical limit of spatial resolution was calculated using the Abbe and Rayleigh criterion of resolution.(10) Table 1 presents the results of spatial resolution limited by temporal and spatial coherence. All the resolution values were predicted using $\lambda = 455nm$, $D = 50\mu m$, $z_1 = 5cm$, and the parameters that specify the USAF-1951 test target: $z_2 = 1.5mm$ and $n = 1.5$. Variation in the spectral width $(\Delta\lambda, FWHM)$ was considered to contemplate the presence and absence of the optical filter, considering the widest spectral width of the LEDs used.

Table 1 – Spatial resolution limited by temporal and spatial coherence for the LHM systems, considering $n = 1.5$, $\lambda = 455nm$, $D = 50\mu m$, $z_1 = 5cm$, and $z_2 = 1.5mm$.

	Theoretical limit of spatial resolution			
	Abbe criterion		Rayleigh criterion	
	Spatial	Temporal	Spatial	Temporal
Δλ = 2 nm	**0.82 µm**	0.76 µm	**1.00 µm**	0.93 µm
Δλ = 30 nm	0.82 µm	**2.92 µm**	1.00 µm	**3.60 µm**

Source: By the author.

As can be observed in Table 1, in the presence of the optical filter ($\Delta\ \lambda = 2nm, FWHM$), the spatial resolution should be limited by spatial coherence, while in the absence of the optical filter ($\Delta\lambda = 30nm, FWHM$), it should be limited by temporal coherence. However, for values smaller than the sensor resolution, which is considered twice the pixel size, (62) this spatial limitation is determined by the digital sensor used.

For the multiheight version of the developed LHM systems, a 3D-printed structure was created to hold all the component devices. Also, an acrylic base ($20cm \times 15cm$) was used as a base for the mentioned 3D structure together with the servomotor that controls the motorized actuator, as shown in Figure 18.

The image acquisition process was developed to be automated by a homemade algorithm, which integrates the control of the actuator and the camera system. The GUI implemented for this software is shown in Figure 19. The Python code for the early version of this software (with the interface presented on the command prompt) is presented in Appendix B. With this software, the acquisition time for a sequence of six holograms with different sensor-sample distances is about $30s$.

The sequence of original holograms is used as input for the digital image process developed especially for this instrumentation. The main steps to perform hologram reconstruction is autofocusing, image alignment, and phase recovery. The Python code developed for this phase recovery (by multiheight method) is presented in Annex A. Using

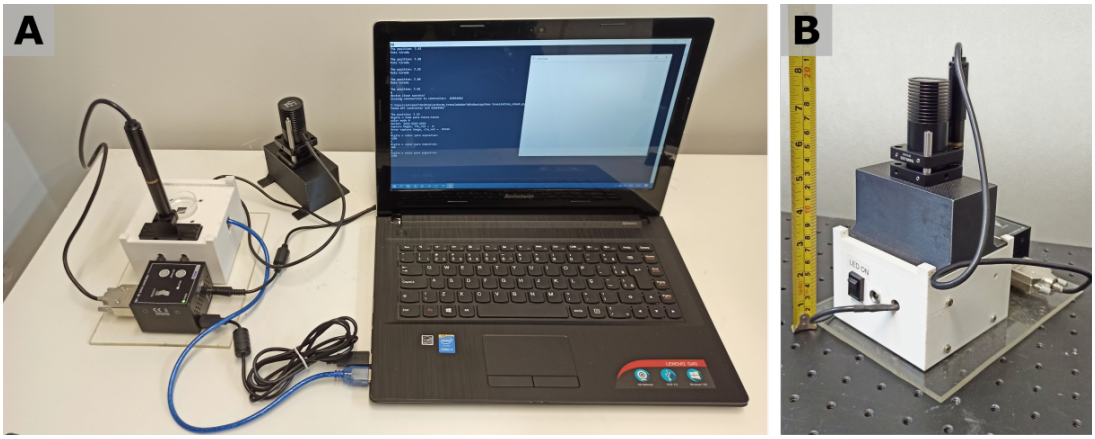

Figure 18 – Multiheight lens-free holographic microscope (a) connect to a laptop computer and (b) the equipment itself.

Source: By the author.

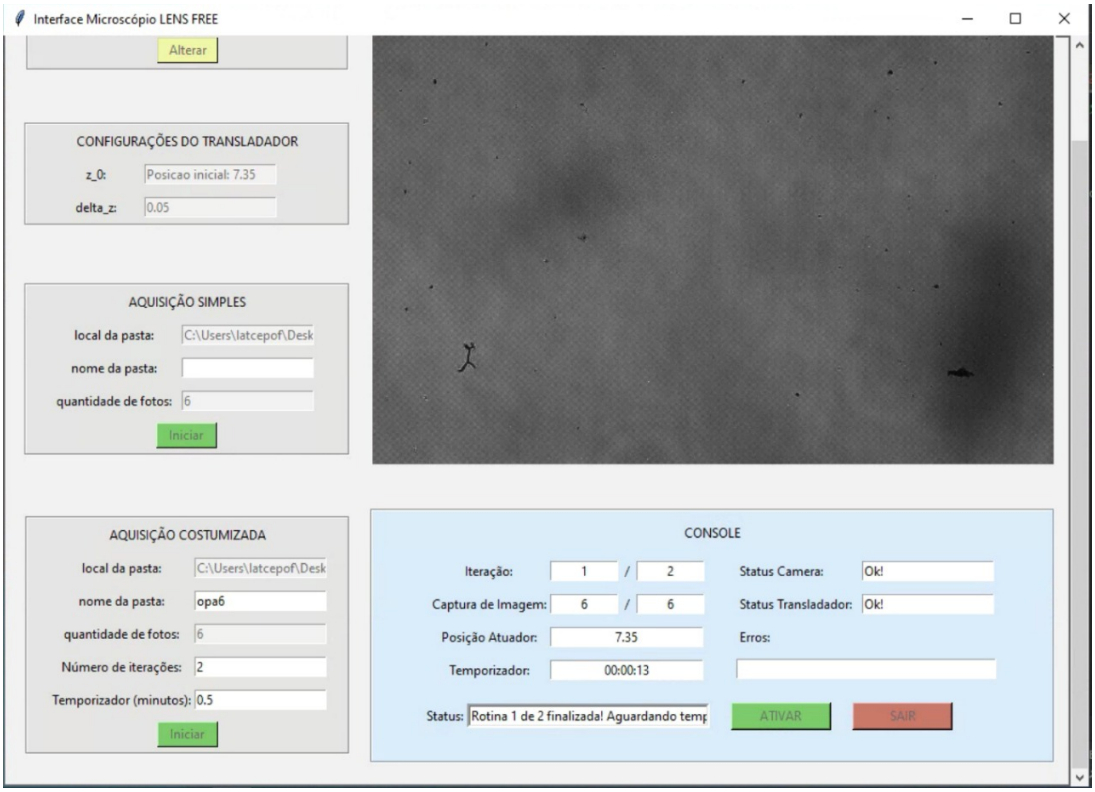

Figure 19 – Graphical user interface developed for the multiheight LHM equipment.

Source: By the author.

a laptop (AMD Ryzen 5 3500U PC, 2.10 GHz, 16.0 GB RAM), the time processing for the hologram reconstruction is about $8min$.

The hologram reconstructed by multiheight and multispectral methods is presented in Figure 20. The smaller elements distinguishable for the image result from this method is the first element of group 8, which has $1.95\mu m$ wide. Therefore, the lateral spatial resolution of this imaging system is about $3.9\mu m$.

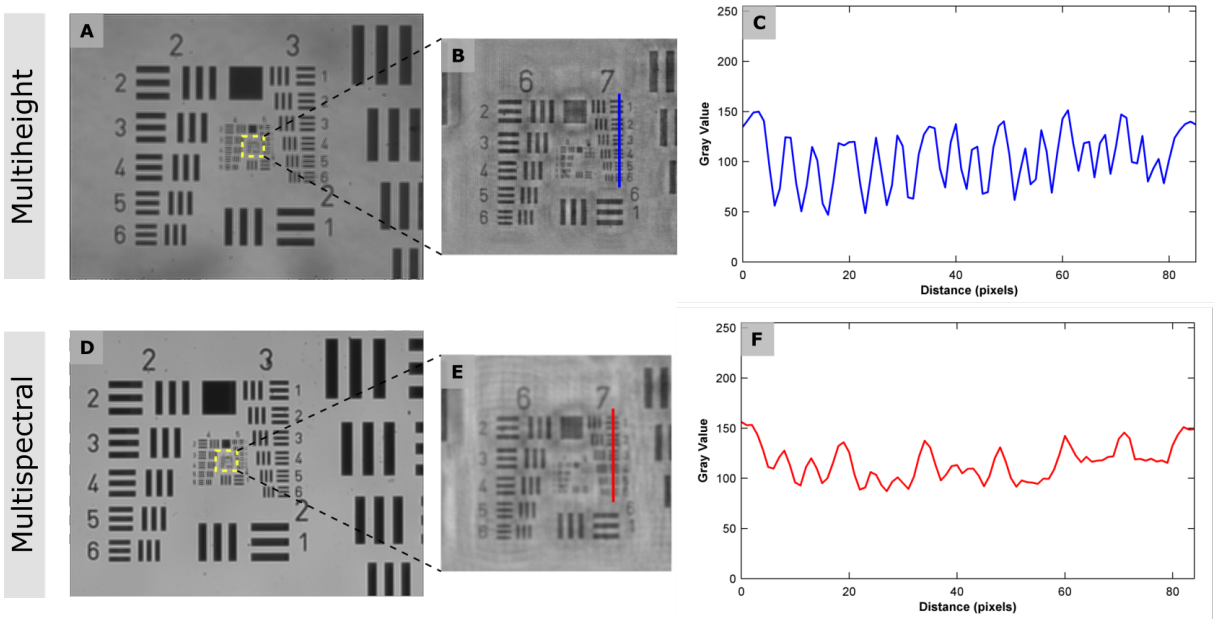

Figure 20 – Reconstructed holograms of the USAF-1951 test target, using the final versions of multigheight and multispectral LHM systems. (a) The entire amplitude image resulted from multiheight method, with (b) elements of groups 6 and 7 amplified and (c) intensity profile of horizontal elements of group 7. (d) The entire amplitude image resulted from multispectral method, with (e) elements of groups 6 and 7 amplified and (f) intensity profile of horizontal elements of group 7.

Source: By the author.

4.3.2 Multispectral

The multispectral lens-free holographic microscope was developed with simple hardware which has a 3D-printed structure to accommodate all the component devices. The entire equipment has $20cm \times 10cm \times 10cm$, as shown in Figure 21.

The addition of an electronic relay on the camera system was an important hardware implementation for this equipment since it prevented the sensor from overheating. An electronic relay was used to control the physical connection of the camera by computational commands. This, added to the homemade RGB lighting system, makes possible the total computational control for this equipment. Therefore, the automation of this system

was performed by the homemade control software. The Python code for this software is presented in Appendix D. With this software, the acquisition of RGB holograms takes less than $30s$. Also, it controls the equipment to perform a sequence of hologram acquisitions in time, to monitor the sample, without user supervision.

The Python code for phase recovery by the multispectral method is presented in Appendix C. With a laptop (AMD Ryzen 5 3500U PC, 2.10 GHz, 16.0 GB RAM) the total time for the phase recovery is about $4min$.

Using the multispectral method for the reconstruction of the USAF-1951 target holograms, the smallest distinguishable element was the sixth element of group 7, which has $3.10\mu m$ wide. Therefore, the lateral spatial resolution of this imaging system is about $6.2\mu m$.

Figure 21 – Multispectral lens-free holographic microscope (a) connect to a laptop computer and (b) the equipment itself.

Source: By the author.

In order to compare the developed LHM systems with the main similar lens-free microscopes in literature, the important parameters of each instrument are presented in table 2. This table shows the similarities and differences between the assemblies, such as the choice of the spatial filter; which can be an optical fiber or a pinhole positioned very close to the light source.

The hardware composition of all the imaging systems in table 2 are similar, although the hologram recording can differ in method and quantity. More complex digital imaging analysis usually requires a larger amount of holograms to be registered, as in the case of super-resolution images. (44) The sum of phase recovery methods (such as multiheight and multispectral) also contributes to increasing holograms requirements and increases the acquisition time.

Another study presents an open-source, cost-effective, portable, 3D-printed LHM

system. (13) In this study, Tobon-Maya and collaborators focus on providing low-cost instrumentation with adjustable settings, without automating the system. The performance of the microscope presented by them depends on the chosen settings: the light source used and the distance established between the sample and the sensor. The reconstruction of these holograms is based on a single image, performing only the numerical propagation of light. The hologram reconstruction presented by their software is available as an ImageJ plugin.

Both lens-free holographic microscopes presented in this study were developed considering the hardware compactness and software that could offer low computational costs and automated systems. Therefore, the final versions of these LHM systems consist of low-cost microscopes with reduced dimensions and no requirement for commercial software to be used. The acquisition time was reduced by the software that performs the device control and also by the reduced number of holograms in the acquisition process. The algorithms were developed in an open-source computational language (Python) to provide phase recovery for both methods (multiheight and multispectral) for a general type of sample and have the advantage to facilitate modifications in order to be complemented with other digital image processing analyses to be developed.

Table 2 – Comparison of in-line digital holographic microscopes with instrumentation similar to the systems developed in this study.

Parameters	**Greenbaum et al., 2012** (44)	**Allier et al., 2017** (45)	**Bian et al., 2018** (46)	**Multiheight LHM system (this study)**	**Multiwavelength LHM system (this study)**
Light source(s)	Xenon lamp + monochromator	LEDs	LEDs	LED + optical filter	RGB LEDs
Spatial filter diameter	$100\mu m$ (optical fiber)	$150\mu m$ (pinhole)	$200\mu m$ (optical fiber)	$50\mu m$	$50\mu m$
λ of the light source(s)	$550nm$	452, 521, and $636nm$	415, 445, 465, 520, 595, 640, and $670nm$	$455nm$	453, 518, and $632nm$
z_1	$10cm$	$5cm$	$7cm$	$5cm$	$5cm$
z_2	$\sim 0.7 - 1mm$	$1 - 2mm$	$\sim 3mm$	$\sim 2mm$	$\sim 2mm$
Pixel size	$2.2\mu m$	$1.67\mu m$	$1.67\mu m$	$1.67\mu m$	$1.67\mu m$
Input images	~ 80	3	35	6	3
Numerical propagation	Angular Spectrum	Fresnel	Angular Spectrum	Angular Spectrum	Angular Spectrum
Phase recovery	Multiheight	Multispectral	Multiheight + multispectral	multiheight	Multispectral
Additional image processing	Super-resolution	Correction of phase values	λ calibration	Autofocus	Autofocus
FOV	$\sim 24mm^2$	$\sim 30mm^2$	$\sim 22mm^2$	$\sim 30mm^2$	$\sim 30mm^2$
Spatial resolution	$< 1\mu m$	$\sim 5\mu m$	$\sim 1\mu m$	$\sim 4\mu m$	$\sim 6\mu m$

Source: By the author.

4.4 Conclusion

Two lens-free holographic microscopes were constructed with hardware and software specially developed for each applied method: multiheight and multispectral. Both systems are portable, lightweight, automated, and have software based on open-source codes.

For both systems, the diameter of the pinhole used and the distance between the light source and the sample were defined based on the results of the initial tests, performed with the multiheight LHM system.

The systems have about $5\mu m$ of spatial resolution ($\sim 4\mu m$ for the multiheight method and $\sim 6\mu m$ for the multispectral) and a wide field of almost $30mm^2$, with represents more than 19 times a FOV of a traditional microscope system with 100x of magnification.

Multiheight LHM equipment has better resolution. The difference, related to the other method, is provided by high temporal coherence. This characteristic is achieved with the addition of the optical filter that reduces the spectral width ($\Delta\lambda$) of illumination.

The multispectral lens-free holographic microscope developed in this study is more compact, lightweight, and low-cost, compared to the multiheight instrumentation. Furthermore, it is totally automated and optimized to avoid sensor heating for more than 2^oC.

Both LHM systems have their software written in Python, which is a friendly and open-source wide-used computational language. The computational processing time for the reconstruction of the holograms by the multiheight method was approximately $8min$. While reconstruction processing by the multispectral method takes about half of the time, as it is simpler.

5 INITIAL APPLICATIONS

In the previous chapter, the lens-free holographic microscopes developed in this study were characterized using a resolution test target. Although this standard sample offers essential information, it poses no challenges to the imaging system beyond the resolution limit. Other samples usually present conditions different from the ideal to be analyzed under the microscope. Lower contrast, motion, and high density are examples of sample characteristics that pose some difficulty to the observation of the sample by lens-free holographic microscopy (LHM).

In order to evaluate the performance of the lens-free microscopes related to these limitations, different biological samples were imaged with multiheight and multispectral LHM systems. These imaging tests were performed throughout the equipment development process. Thus not all the results presented were obtained with the final version of each lens-free microscope. For this reason, the equipment details are specified for each case.

Considering the great potential of LHM for diverse applications, some samples were chosen to be imaged. Each section of this chapter presents one of these applications, providing their importance and the difficulties and facilities related to their use in the LHM systems.

5.1 Cell cultures

5.1.1 Introduction

2D cell culture is widely used for *in-vitro* investigations on biological systems because it provides a cell culture monolayer with controlled parameters. (65) Considering the relevance of this biological sample for scientific investigations, we evaluated the potential use of multiheight and multispectral LHM systems for *in-vitro* experiments with cells, using two different cell lines (healthy and carcinogenic).

Intrinsic characteristics of the cells, such as their morphology, and their distribution can influence the quality of the image. In a 2D cell culture, the cell distribution is directly related to the density of the cells in the dish, which is called confluence. Although cell size usually is dozens of micrometers, the cell culture confluence can limit the applications of imaging systems with few micrometers of spatial resolution, as the LHM systems presented in this study.

Cells usually have low light absorption. Therefore, to be imaged under transmission images, they require staining to have an appropriate contrast. Since the staining is toxic for the cells, living cells are usually observed under phase contrast. Considering that lens-

free holographic images provide amplitude and phase images of the same observed scene, it can be an advantage for this imaging system. Additionally, to facilitate cell culture visualization under an LHM system, cells can be cultivated in cell culture dishes specific to be used under microscopy.

5.1.2 Material and Methods

5.1.2.1 Cell culture

Human dermal fibroblasts neonatal (HDFn) and squamous cell carcinoma SCC-25 (American Type Culture Collection - ATCC, Wesel, Germany), were cultivated at 37^oC in humidified 5% CO_2 atmosphere in Dulbecco's modified Eagle medium (DMEM) supplemented with 10% (v/v) Fetal Bovine Serum (Cultilab, Brazil). Cells were plated on a 35 mm glass bottom dish (Greiner Bio-One, Germany) at a density of 5×10^5 cells and let in the incubator (37^oC, 5%, CO_2) overnight.

HDFn was also cultivated in DMEM without phenol red, supplemented with 10% (v/v) Fetal Bovine Serum (Cultilab, Brazil). The presence of phenol red, a pH indicator, directly influences the absorption of the media and, consequently, changes its observed color, as shown in Figure 22.

5.1.2.2 LHM imaging

Cell cultures were imaged by multiheight and multispectral lens-free holographic microscopes. As mentioned in Chapter 4, both imaging systems record holograms by transmission, using a partially coherent illumination. The light source is positioned $5cm$ above the cell culture dish, which is directly positioned over the digital image sensor (MT9J003, Aptina/Onsemi, USA).

HDFn and SCC-25 cell cultures imaging were performed by early versions of both lens-free holographic microscopes. Thus, different from the final version, the light source of a multispectral LHM system was an RGB lamp (7,5 W, A60 RGB, Osram, Germany), with emission centered at 634 nm, 518 nm, and 462 nm, with a spectral width of 21 nm, 33 nm, and 23 nm, respectively. Additionally, for both systems, the light was spatially filtered by a $150\mu m$ diameter pinhole (P150S, Thorlabs, USA).

The HDFn cells were also imaged with the final version of the multiheight LHM system. Both versions of multiheight lens-free microscope used a blue LED, with emission at $455nm$, but the last version filtered its light with a $50\mu m$ diameter pinhole (P50S, Thorlabs, USA) and an optical filter with $2nm$ of spectral width (LL01-442-12.5, Semrock, USA).

The cell culture imaging was performed out of the cell incubator, at room temperature.

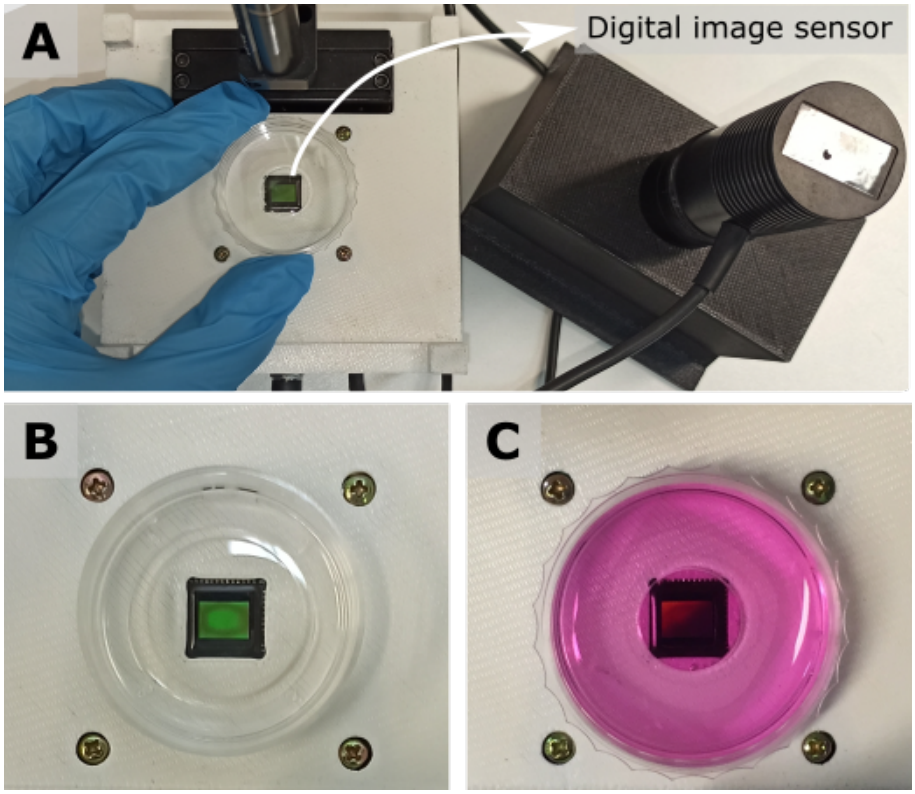

Figure 22 – Upper view images of cell culture on the lens-free holographic microscope (a) with the sample being positioned on the digital image sensor of the LHM system and (b) the cell culture dish with DMEM medium without phenol red and (c) DMEM medium with phenol red.

Source: By the author.

5.1.3 Results and Discussion

As the cell cultures were imaged out of their incubator, the temperature variation between the inside and outside of the incubator causes condensation on the cell dish cover. To avoid external debris, the cell dish cover was removed. Although it improved the quality of the recorded holograms, this situation is not ideal for cell culture, as it exposes the sample to contamination.

Both observed cell types are approximately $20\mu m$ in their largest dimensions. HDFn cells are spindle-shaped,(66) thus they have elongated shapes.(67) SCC-25 are epithelial-like cells,(68) then, different from the HDFn, they are polygonal in shape with more regular dimensions.(67) These described morphologies for HDFn and SCC-25 cells could be observed by LHM systems, even by their first versions, as shown in Figure 23 and 24, respectively. These reconstructed holograms were obtained from images recorded with a lower coherence light, compared with the images of the final versions, because the light was only filtered by a $150\mu m$ diameter pinhole.

As HDFn is refractile cells,(66) they can be observed with a great contrast under

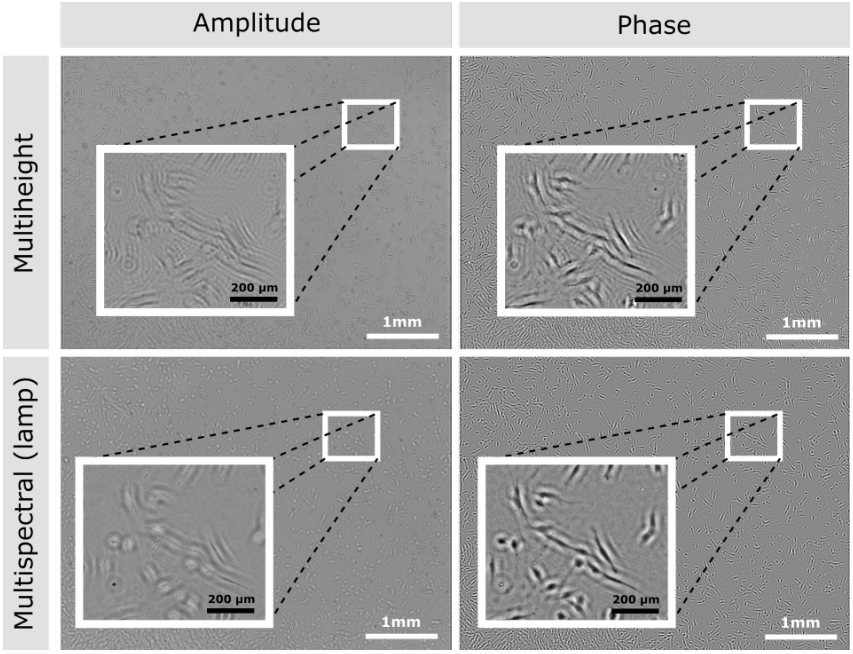

Figure 23 – Amplitude and phase images of HDFn cell culture, resulting from hologram reconstruction using multiheight and multispectral methods, with a pinhole of 150μm in diameter in both systems and the RGB lamp in the multispectral LHM system.

Source: By the author.

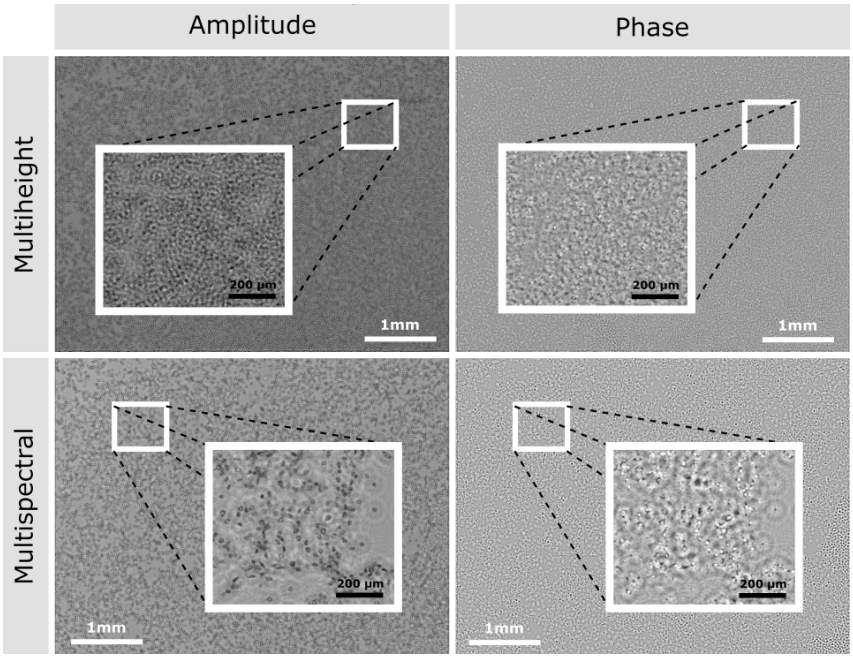

Figure 24 – Amplitude and phase images of SCC-25 cell culture, resulting from hologram reconstruction using multiheight and multispectral methods, with a pinhole of 150μm in diameter in both systems and the RGB lamp in the multispectral LHM system.

Source: By the author.

phase images. This characteristic was evidenced when this cell type was observed under lens-free microscopes. As can be seen in Figure 23, the phase images show HDFn cells' boundaries more clearly. For SCC-25-reconstructed holograms, in Figure 24, amplitude images provide better results, with higher contrast between cells and background.

Images from the multiheight method have more artifacts compared to images from the multispectral method. This is probably due to the fact that the hologram acquisition time is longer for the multiheight method. Therefore, there is a greater chance that the alignment process was not sufficient to suppress the effects of sample movement in these cases.

Figure 25 presents the phase and amplitude images from the final version of multiheight lens-free microscope, recorded with the use of a smaller pinhole (with $50\mu m$ in diameter). As this spatial aperture reduces, the coherence of the light increases and provides a better resolution. Therefore, compared to the images in Figure 23, more details from cells can be observed. Figure 25 also shows that the presence of phenol red in the cell culture medium contributes to a slighter attenuation of the light that reaches the image sensor. This effect is more evident in the phase image because it presents more sample details than the amplitude image.

For all cell cultures observed, regions with low confluence provided reconstructed holograms that present morphological information about the observed cells, with great contrast and resolution for one of the resulting (amplitude or phase) images. In this case, cell distribution and size can be quantified from these images. Nonetheless, high-confluence regions, where the cells are very close to each other, do not provide clear information about these individual microorganisms.

5.2 Blood smear

5.2.1 Introduction

The blood smear is a traditional sample used for medical analysis. (69) It usually is stained and observed by specialists using a bright-field optical microscope. Morphological and counting information about cells and their internal structures are essential for this analysis.

Different types of cells are presented in a blood smear and each one has its own morphology: red blood cells, white blood cells, and platelets. Red blood cells (erythrocytes) are the most abundant cell in the blood. They have about $7\mu m$ size and a discoid (bi-concave) shape. (70) White blood cells (leukocytes) have larger dimensions compared with red blood cells, about $15\mu m$, and are much less common than erythrocytes. (71) The platelets (thrombocytes) are the smallest cell elements of the blood, with about $3\mu m$, because they are in fact cytoplasmic fragments from large cells.

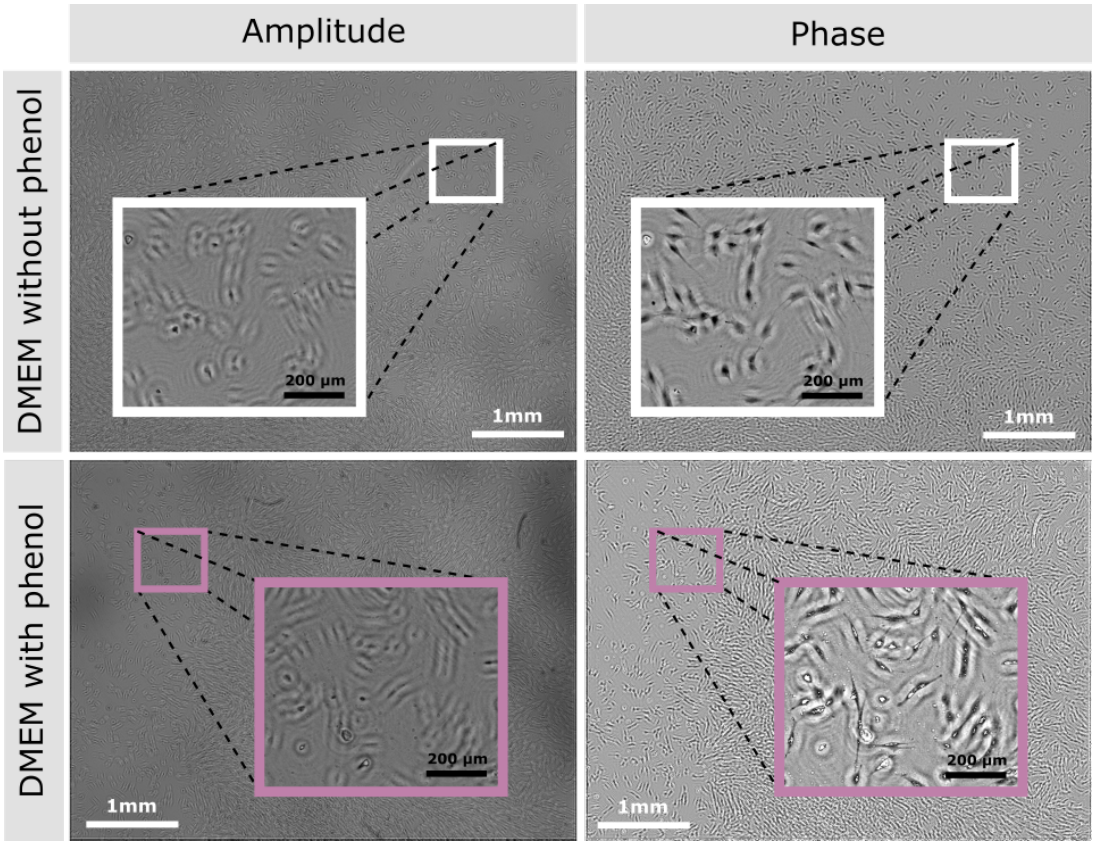

Figure 25 – Amplitude and phase images of HDFn cell culture, prepared with DMEM culture medium with and without phenol red, resulting from hologram reconstruction using multiheight method, with a pinhole of $50\mu m$ in diameter.

Source: By the author.

Blood smears and other types of blood samples have been analyzed by other researchers using lens-free microscopes. Most of them used shadow images with incoherent illumination (72) or explored the diffraction pattern from the holograms (52, 73–75) to classify and count erythrocytes or leukocytes and their subtypes. Although analysis of these pattern images provides accurate results for low-density samples, reconstructed images from holograms of blood cells perform better counting in higher concentrations. (76)

For the presented lens-free microscope, the blood smear imaging represents a performance test to observe the reconstructed image quality. It also represents a resolution challenge for this instrumentation, once the size of the cells approaches the spatial resolution of the presented imaging systems. Traditional methods for sample preparation use air-dried cells on glass slides. Although the cells remain static, their density can influence the quality of these observed images.

5.2.2 Material and Methods

5.2.2.1 Blood smear

The blood smear used in this study was previously prepared by specialists, to be observed by traditional methods in other studies.(77) This sample basically consists of a drop of blood spread and air-dried on a glass slide. This blood smear was stained with May Grünwald-Giemsa, which is a combination of dyes that promote visualization of different cellular structures under light absorption contrast.

5.2.2.2 LHM imaging

The blood smear holograms were performed with the final version of the multi-height LHM equipment. The sequence of holograms (recorded at different sample-sensor distances) is obtained using an LED with emission centered at 455 nm, which is spatially filtered by a $50\mu m$ diameter pinhole (P50S, Thorlabs, USA) and spectrally filtered by an optical filter with $2nm$ of spectral width (LL01-442-12.5, Semrock, USA).

5.2.3 Results and Discussion

Blood smear reconstructed holograms are presented in Figure 26. In both (amplitude and phase) images there is enough contrast to differentiate the sample from the background. Nevertheless, it is hard to identify single cells, even in the feather edge of the smear, which is the furthest region from the blood drop application point, where the sample is sparse. As the proximity of the cells increases, their separation becomes impracticable by these LHM images. These reconstructed images from the recorded holograms do not have sufficient resolution for the traditional analysis of blood cells or their internal structures.

Considering these issues, different approaches should be used to analyze blood smear images. LHM systems that apply super-resolution methods can be used to observe this type of sample with the traditional preparation, as presented by Roy et al. (35) Furthermore, blood cells can also be observed by simpler approaches, as long as they are properly prepared in dilutions, in order to have each cell shadow or hologram patterns analyzed individually. (52, 72–75)

5.3 Microalgae

5.3.1 Introduction

As another example of biosample, microalgae were also imaged by the presented lens-free microscopes. Microalgae are unicellular photosynthetic micro-organisms, living in saline or freshwater environments.(78) These microorganisms have drawn significant interest worldwide due to their great potential for applications. They have been applied

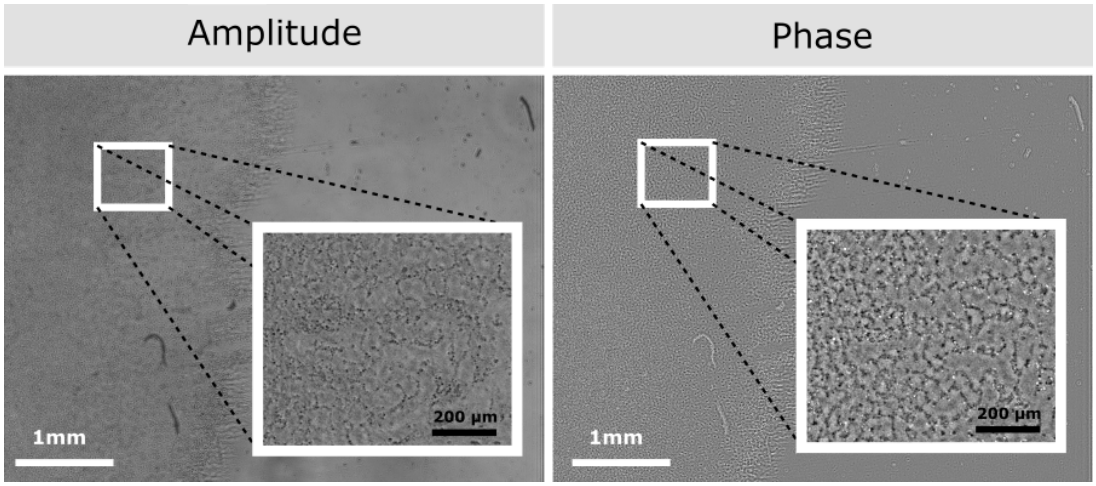

Figure 26 – Amplitude and phase images of a blood smear, resulting from hologram reconstruction using multiheight method, with a pinhole of $50\mu m$ in diameter.

Source: By the author.

in the synthesis of biopolymers, to remove pollutants from wastewater, and also in the green energy, pharmaceutical, nutraceutical, and cosmetic industries. (79–82)

Although LHM systems have been used to investigate microalgae cells, most of them are not exploring the reconstructing images from holograms but extracting and analyzing features of the hologram pattern itself. Researchers have explored the fingerprint shadow patterns of microalgae species to observe flocculation,(83) morphological changes (84), and to judge the liveness and death of cells. (85)

Under the LHM system, microalgae samples can be imaged on the solution and do not present significant movements in the time scale of the performed image recordings.

5.3.2 Material and Methods

5.3.2.1 Microalgae

Two different species of microalgae were provided by the Hydraulics Laboratory of São Carlos School of Engineering (EESC): *Desmodesmus sp.* and *Dolichospermum flosaquae*. These microorganisms were maintained in an aqueous medium.

5.3.2.2 Microscopy imaging

For each species, a small drop of the sample was placed between a glass slide and a coverslip, immediately before being observed under microscopy.

The microalgae holograms were performed with the final version of the multiheight LHM equipment, using a blue LED ($455nm$) filtered by a $50\mu m$ diameter pinhole (P50S, Thorlabs, USA) and an optical filter with $2nm$ of spectral width (LL01-442-12.5, Semrock, USA).

In order to compare the hologram reconstruction images, transmission images were obtained from a traditional optical microscope (Eclipse-S, Nikon, Japan), using 40× and 10× objective lenses, respectively for *Desmodesmus sp.* and *Dolichospermum flosaquae* samples.

5.3.3 Results

Each one of the observed species has its own morphology. Thus, their imaging provides different tests for lens-free holographic microscopes.

In general, *Desmodesmus* species are round-shaped, non-motile, single-celled coccoid microalgae lacking flagellae and rigid shells, and have about $2-8\mu m$ in diameter.(86) They are usually organized in cenobium form, with about three cells arranged linearly, as pointed out by the white arrows in the images from traditional microscopy in Figure 27.

The *Desmodesmus sp.* size is close to the presented LHM resolution. Also, their group organization provided an additional challenge to identify the microorganisms individually in the holographic images. It can be observed in the reconstructed holograms in the first row of images of Figure 27, in which the amplitude and phase from *Desmodesmus sp.* LHM images are shown.

Dolichospermum flosaquae are planktic round cells (87) that compose filamentous elements. Because they have longer structures, they ate observed more clearly than *Desmodesmus sp.*, as can be observed in the bottom row of Figure 27, in which are presented amplitude and phase images of the *Dolichospermum flosaquae* sample.

Despite the low level of detail presented in these images, LHM systems can be used to obtain reconstructed images with a high level of detail, since it applies specific tools to achieve this goal. For example, Göröcs *et. al* (47) presented high-resolution color reconstruction images provided by an LHM flow imaging system that applies learning-based phase-recovery developed especially to analyze microalgae.

Additionally, raw hologram images can be used to have features extracted from their patterns in order to offer other analyses. Thus, analysis beyond the reconstructed image details can be provided, even considering morphological aspects.(83–85)

5.4 *Ceriodaphnia*

5.4.1 Introduction

In addition to the micrometer-scale samples mentioned above, the LHM instrumentation was also used to detect *Ceriodaphnia*, which are microorganisms of about $1mm$ that belong to the order Cladocera. Cladocerans are microscopic aquatic Crustacea, which colonize almost every type of freshwater and are excellent indicators of environmental change. For this reason, they are usually applied for toxicological tests. (88) Mortality, re-

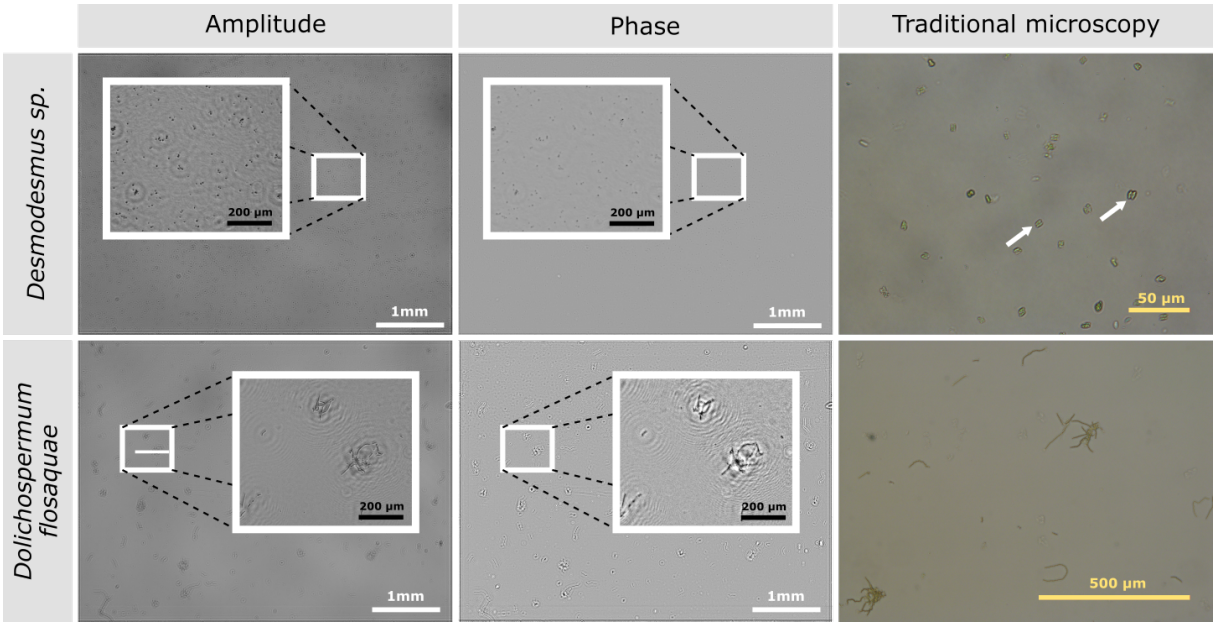

Figure 27 – Amplitude and phase images of the microalgae species:*Desmodesmus sp.* and *Dolichospermum flosaquae*, resulting from hologram reconstruction using multiheight method, with a pinhole of $50\mu m$ in diameter and an optical filter with $2nm$ of spectral width. Another imaging area from traditional bright field microscopy is present for both species on the right column. The upper image was performed with $10\times$ objective lens and the bottom image with $4\times$ objective lens.

Source: By the author.

production, and growth rate are parameters for evaluating test chemicals' toxicities using Cladocerans. (88)

Sunayama and collaborators developed an assembly based on low-coherence digital holography for the size measurement of freely-swimming Daphnia (another genus of Cladocera order). (89) They used low coherence light to reduce speckle noise and the advantage of digital post-focusing to imaging moving targets.

Lens-free holographic microscope presents a promising alternative to detecting these microorganisms since it is portable and low-cost equipment. As these living microorganisms move very quickly in water, this detection represents a challenge for the presented microscopy instrumentation.

5.4.2 Material and Methods

5.4.2.1 *Ceriodaphnia*

Living *Ceriodaphnia sp.* were provided in water by the Aquatic Limnology and Ecotoxicology Laboratory of the Federal University of São Carlos.

5.4.2.2 LHM imaging

The *Ceriodaphnia* holograms were performed with the early version of the multi-height LHM equipment, with pinhole $150\mu m$ in diameter (P150S, Thorlabs, USA), using an optical filter with $2nm$ of spectral width (LL01-442-12.5, Semrock, USA).

Once there is a limitation due to the movement of this microorganism, causing distortion on the majority of recorded microorganisms images, only one hologram was recorded for each sample.

To provide more visual details from the sample, transmission images were obtained from a traditional optical microscope (Eclipse-S, Nikon, Japan), using $40\times$ and $10\times$ objective lenses.

5.4.3 Results and Discussion

A single recorded hologram from *Ceriodaphnia* contains sufficient information to provide focused images, but not to perform the phase recovery calculation. These optical transmission holograms just offer contour details of each imaging *Ceriodaphnia*, as shown in Figure 28. This is sufficient to measure the size of each one of these microorganisms, which provides the quantification of the growth rate.

The detection of these cladocerans by the developed lens-free microscope is possible. However, there are some limitations inherent to this used setup that makes it difficult to obtain better results. Due to the fast movement of *Ceriodaphnia*, the major limitation is the low rate of image capture of the camera (approximately $2fps$). It caused a distortion in the images of moving microorganisms, which invalidated most of the recorded images.

Considering the imaging of living cladocerans, the post-focusing function, inherent in LHM, is an advantage for this application. For optimization of the analysis of these microorganisms, a digital camera with a high frame rate is required.

Although LHM images can offer some morphological information about this species, it is important to note that there is a great similarity between the microorganisms of the order Cladocera, so the identification of each species can only be made by applying more detailed analyzes offered by other techniques. (90)

5.5 Yeast cells

5.5.1 Introduction

Another sample observed under the presented LHM system was yeast cells, which have an essential role in brewing, baking, and biofuel production. The most common method to perform yeast cell detection and counting is using a hemocytometer together with a bench-top optical microscope to visually identify cells. Besides applying bulk and

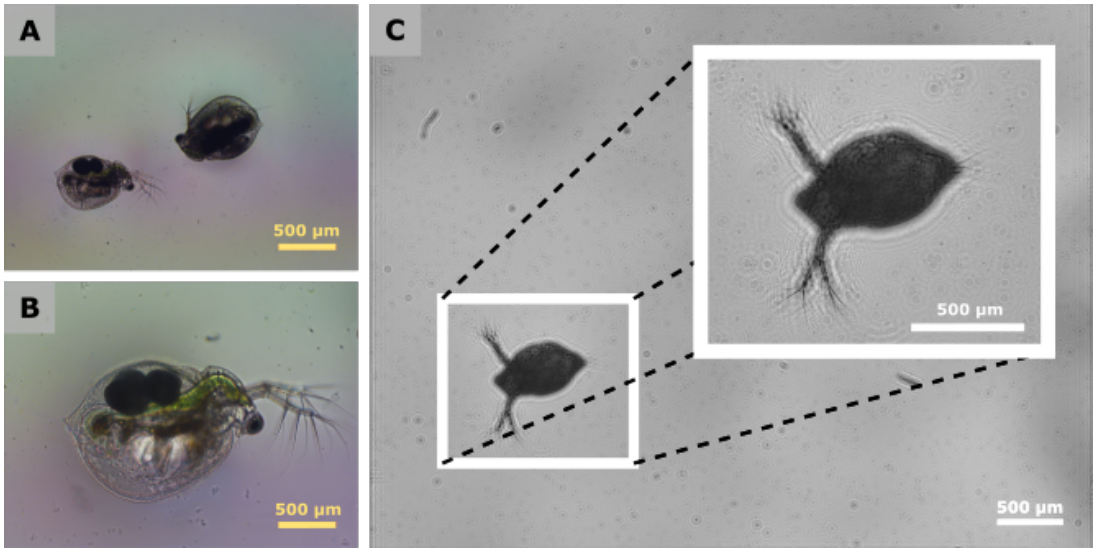

Figure 28 – Ceriodaphnias microscope images (a) from traditional bright field optical microscope with 100× of magnification (using a 10× objective lens) and (b) with 400× of magnification (using a 40× objective lens) and (c) the in-focus hologram from the LHM system.

Source: By the author.

costly instrumentation, it used to be very time-consuming. A flow cytometer is an automated but expensive alternative. Considering this scenario, Feizi *et. al* proposed an LHM and machine learning algorithms to provide viability and concentration analysis for yeast cells. (60)

In this presented study, yeast cells are used to perform quantitative analyses for the LHM system. Yeast cells in solution were chosen for this type of analysis because of their low cost, easy acquisition, and uniformity in their microscopic elements (cells). Thus, lens-free holograms were used here to detect and count different concentrations of yeast solutions with the purpose of testing the developed instrumentation and the counting algorithm.

In order to compare and validate the presented results, the detection and counting of yeast cells in solution were performed automatically by LHM and manually using a hemocytometer (or Neubauer chamber) with traditional optical microscopy.

5.5.2 Material and Methods

5.5.2.1 Yeast solution

The yeast solutions were prepared with *Saccharomyces cerevisiae*, using commercial dry yeast (W34/70, Fermentis, France).

Firstly, dry yeast was rehydrated in distilled water, as presented by Feizi *et. al.*. (60) But due to a high degree of flocculation, manual and automatic counting became

difficult and it add errors to each one of these quantification methods. Therefore, to avoid flocculation, the solutions were prepared using the culture medium YPD (Yeast extract-Peptone-Dextrose), which is composed of 1% yeast extract, 2% Dextrose (D-glucose), and 2% of peptone. The stock solution was prepared with 0.02 g of dry yeast and 10 mL of YPD and was maintained on $37oC$ in humidified 5% CO_2 incubator for 2h. The first solution was prepared with 1:10 of stock solution. Other dilutions were prepared using the proportions 1:1, 1:5, and 1:10 of the first solution.

5.5.2.2 Traditional imaging and counting

For the traditional method, using the hemocytometer (or Neubauer chamber), $10\mu L$ was used to fill the chamber. For counting, it was considering 5 central quadrants of the chamber, as shown in Figure 29 (four quadrants from the corners and the central one). The count was performed by summing all the yeast cells within each of these quadrants and half of all the cells quantity that was on the edge of each one. Therefore, the concentration is obtained by dividing the total sum of microorganisms by the volume of the area referring to these 5 quadrants. (91)

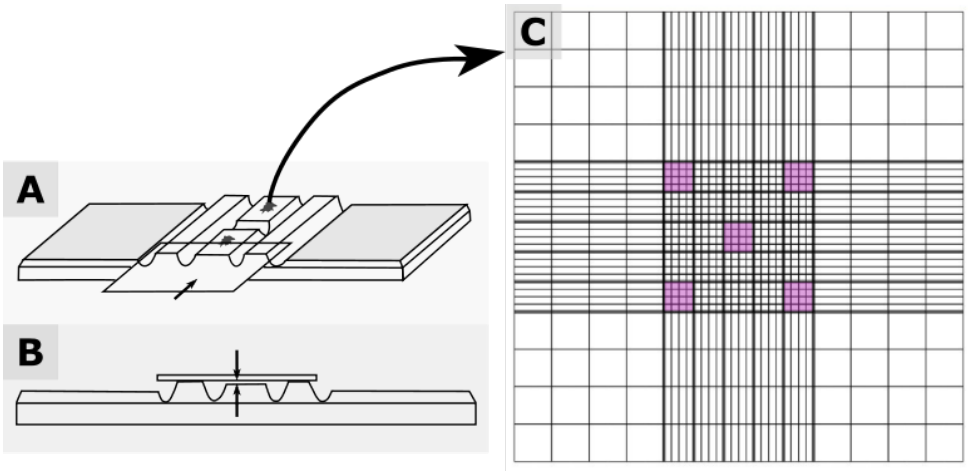

Figure 29 – Schematic of a hemocytometer (or Neubauer chamber) used to perform yeast cells' manual counting. (a) Top view and (b) lateral view of the chamber and its (c) reference quadrants present in the imaged area. The pink quadrants represent the area used to perform yeast counting.

Source: By the author.

5.5.2.3 LHM imaging and counting

The samples for LHM imaging were prepared using a homemade chamber, which is schematically represented in Figure 30. It is composed of a glass slide glued on a coverslip in which holes of $1cm$ in diameter were made. Therefore, the wells where the sample will be placed had the height of the coverslip ($0.2mm$) and the diameter of each hole ($1cm$).

After filling the well with a yeast solution, a squared coverslip was positioned over the well almost covering it entirely. Thus, a small drop of water is placed between the coverslip and the top of the chamber for better adhesion between them. After the well is filled, the coverslip that is over it is slid in order to close the well. Ideally, there should be no bubbles in the well.

The yeast cells holograms were performed with the final version of the multispectral LHM equipment, using RGB LED illumination, pinhole $50\mu m$ in diameter (P50S, Thorlabs, USA), and an optical filter with $2nm$ of spectral width (LL01-442-12.5, Semrock, USA), as represented in Figure 30.

The automated counting was performed by a homemade algorithm developed in Python language using traditional digital image processing operations, which does not involve any machine learning. Amplitude images obtained by multispectral phase recovery by a homemade algorithm, also developed in Python, were used to perform cell detection and counting. To perform yeast cell detection, the first step was to enhance the image contrast using a gamma correction (with gamma factor = 0.5). (92) Then, an intensity threshold converted the grayscale image into a binary image, (92) resulting in a spatial mask with the cells' candidate areas in white and the back of the ground in black. Next, the candidate elements to be considered yeast cells were defined from the detection of the contours of the mask. The detection of each element was followed by the area calculation of that element. In this way, only elements with an area between 2 and 20-pixel sizes were detected as cysts. Finally, the algorithm counted all elements detected as yeast cells in the entire image.

All these parameters mentioned above were defined experimentally by the user, who could observe the real-time result applied to a small area of the hologram, as shown in Figure 30. Once the parameters were chosen, they were used to apply processing to the entire image.

5.5.3 Results and Discussion

Saccharomyces cerevisiae usually have a simple ellipsoidal shape (93), which makes it an uniform cell. Despite its reduced size (about $4\mu m$) regarding the spatial resolution of the presented LHM system, it can be detected when is not presenting flocculation and its round morphology facilitates the application of this tested analysis.

Figure 31.a and 31.b show amplitude and phase images of the sample with the highest concentration of yeast cells, observed from the LHM system using the multispectral method.

Figure 31.c presents the program windows from the homemade software developed to perform yeast cell detection and counting. The window on the right side of the Figure

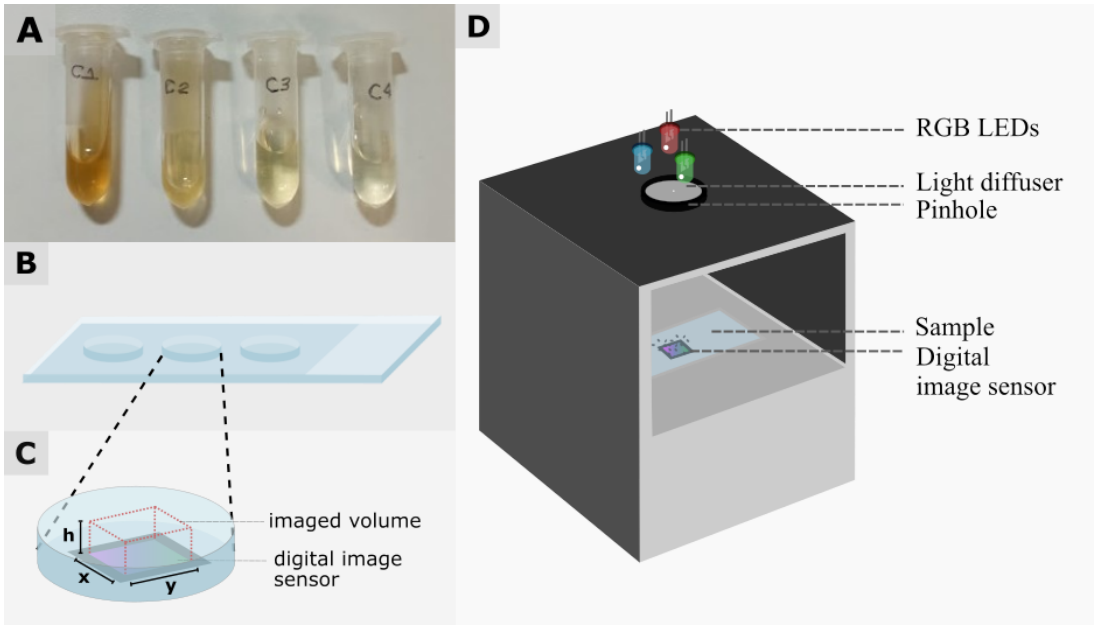

Figure 30 – Representation of yeast cells imaging by LHM. (a) Eppendorf with different concentrations of *Saccharomyces cerevisiae*. (b) A representation of the homemade chamber was prepared to image yeast cells in the LHM system. (c) Imaging volume of each recorded hologram from yeast samples on chamber presented in *b*. (d) Multispectral LHM system with which the yeast holograms were recorded.

Source: By the author.

shows the entire amplitude image to be analyzed. The window on the left side shows the slide bars used to define each parameter of the detection algorithm (size of the blur kernel, intensity threshold, and area values) together with an image cropped from the entire (selected by the user). As the values of each parameter were changed in the slide bars, the detection result was updated and presented in this cropped image.

The graph in Figure 32 shows the results of yeast counting for the samples analyzed manually and automated, using a bright field optical microscope and the multispectral lens-free holographic microscope. The linear equation fitting evidences the agreement between data from manual and automated counting.

The yeast cell concentrations of the sample were defined based on the application range of each technique. Lower concentrations could be quantified by the LHM system and slightly higher concentrations could be quantified by manual counting.

5.6 Conclusion

Both multiheight and multispectral LHM systems were successfully applied to imaging cell cultures, microalgae, and yeast solutions. They can also be used to perform images of living *Ceriodaphnias* with some restrictions, which can be mitigated using an

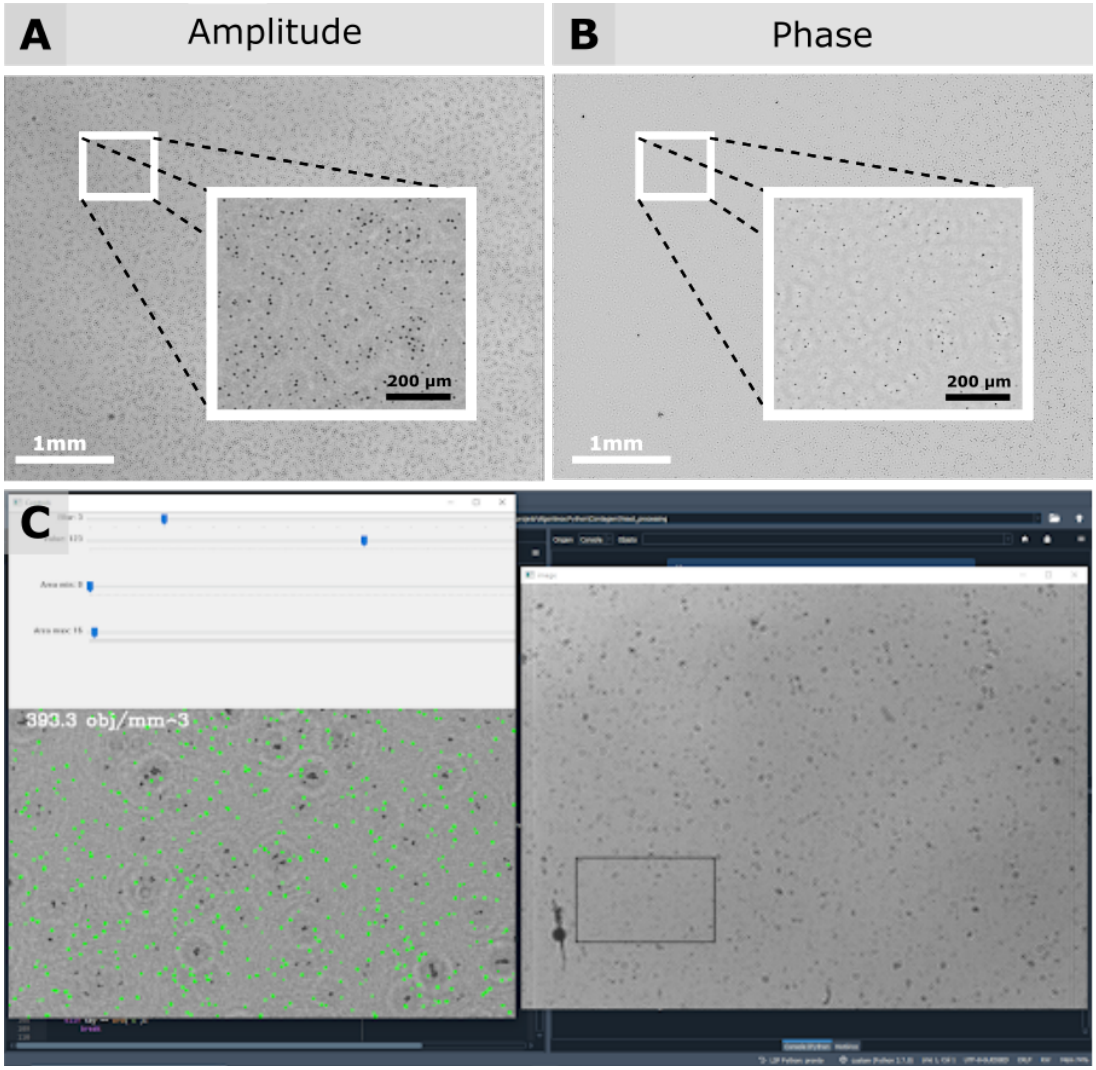

Figure 31 – Yeast cells images and counting from reconstructed holograms. (a) amplitude and (b) phase image related to the highest concentration sample. (C) Image processing windows from the homemade software that performs the yeast cells detection and counting using amplitude images from the LHM system.

Source: By the author.

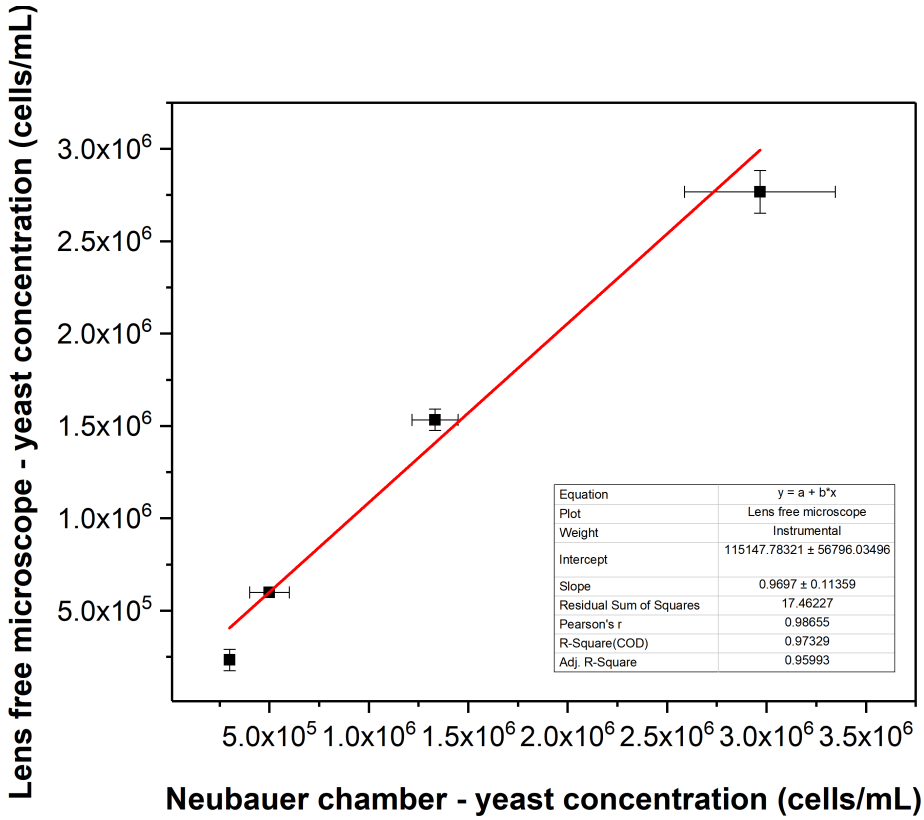

Figure 32 – Graphic of yeast cell concentration obtained by the automated counting performed with holograms from LHM and by manual counting performed with bright field microscopic images using the Neubauer chamber.

Source: By the author.

image sensor with a higher frame rate. Nonetheless, their resolution was not sufficient to provide satisfactory images of a blood smear to be analyzed.

These tested LHM systems can offer more information about microscale sparse samples, such as the used solutions of yeast cells. High confluence cell cultures or high cell density in a blood smear presented a limitation of this imaging technique. To be observed adequately by LHM, these types of samples require super-resolution methods, and consequently more complex hardware and software) and/or specific analysis based on deep learning, which needs a very large database.

Slight movements that cannot be corrected by alignment processing reduce the quality of the resulting image. Considering this issue, the multispectral LHM system has an advantage over the multiheight, because it requires a short time to record all the holograms. Nonetheless, for static samples that require more details, the multiheight lens-free holographic microscope can be used to offer more detailed images.

The developed imaging systems also presented the convenience to have their digital image processing complemented to offer a more complete analysis of the imaged sample,

as shown in the yeast count in the solution.

Both lens-free holographic microscopes presented the great potential to be used for different types of biological samples. Thus, these applications can explore the amplitude and phase information in a wide field of view offered by these systems.

6 DETECTING AND COUNTING OF *GIARDIA* CYSTS

6.1 Introduction

Giardiasis is a parasitic infection responsible for mortality and morbidity, mainly in developing regions. It is included in the "neglected diseases" list, as it is associated with low family income, and lack of water and sanitation infrastructure. (94–96) *Giardia* cysts, i.e., the infective form of this gastrointestinal parasite, range from 7 to 14 μm and present particular environmental and epidemiological relevance due to their low infectious dose in humans (from approximately 1 to 10 organisms), and high survival rate, (97,98) which makes monitoring an important matter.

In water and wastewater samples, Methods 1623.1 and 1693, respectively, are the most common protocols for recovery and quantification. (99,100) Both rely on immunofluorescence assays (IFA) for cyst visualization, in order to distinguish between any debris likely to be present in concentrated samples. In this detection protocol, cysts are labeled with specific antibodies coupled to fluorescein isothiocyanate, then visualized under fluorescence microscopy. (101) IFA leads to a more clear cyst detection by microscopists, as the parasites are expected to present high contrast against the background. (102) However, this is an expensive technique that requires bulky laboratory infrastructure and depends on trained experts for slide preparation, which is also time-consuming in terms of cyst enumeration.(102, 103)

The cons associated with the standard protocols for cyst recovery and visualization encourage research on alternative methods, (104) particularly because these may contribute to studies on protozoan outbreaks and giardiasis epidemiology. (105) In terms of detection approaches, only a few alternative techniques have been explored, for example, performing fluorescence (106) and bright field (107) images from smartphone-coupled devices, and images from an agitated suspension with a developed *in-situ* microscope. (103)

Lens-free holographic microscopy (LHM) is an alternative to bulky optical microscopy that offers a portable, lightweight, and low-cost device. LHM provides phase and amplitude images from holograms recorded with partially coherent illumination. It has been applied to air quality monitoring, (108) cell and tissue visualization, (27) cell death investigation, (61) and cell culture monitoring, (109) for example.

Considering the limitations of usual techniques and those inherent to alternative methods, our study aims to elucidate the potentials of LHM and therefore contribute to label-free, low-cost, and easy-to-run approaches to *Giardia* cysts detection and counting.

6.2 Material and Methods

6.2.1 Target organism

Purified suspensions of *Giardia duodenalis* cysts (H3 isolate) obtained from experimentally infected gerbils were purchased from a commercial supplier (Waterborne, USA). According to the supplier, cysts were purified from feces by density gradient using sucrose and Percoll, and suspensions were provided in an aqueous solution containing PBS with penicillin, streptomycin, gentamicin, and 0.01 % Tween® 20. This purified suspension was kindly provided by professor Lyda Patricia Sabogal Paz, from the Hydraulics Laboratory of São Carlos School of Engineering (EESC).

6.2.2 IFA standard enumeration

The concentration of cysts in our working suspension was determined by standard fluorescence microscopy prior to the detection tests with the lens-free microscope. Although there was an estimate of the number of cysts provided by the supplier (approximately 5×10^6 cysts in $8mL$), cyst counting was performed by IFA in order to follow the standard detection protocol, as well as to provide background for discussing variability associated with this quantification method.

Cyst enumeration was carried out in three different sample volumes spiked directly onto treated slides (carried out in two genuine replicates) and blindly examined by two microscopists (n = 4 enumerations for each sample volume). Selected volumes were 5, 10, and $15\mu L$, chosen for laboratory convenience considering the intended comparison to the lens-free technique.

The stock suspension was homogenized in the vortex mixer for $60s$ prior to being spiked onto microscopy wells and kept drying at room temperature without chemical fixing. The average time for sample drying was $30min$. Cyst enumeration was performed using a commercial detection kit (Merifluor® G/C, Meridian Biosciences, USA), also provided by professor Lyda. In short, the protocol includes staining with the detection reagent and an eriochrome black counterstain and incubating stained samples at 37^oC for $30min$. Excess is rinsed with a buffer washing solution and samples are fixed with a glycerol-based mounting medium, both provided by the manufacturer. This process was carried out under no direct light, and a dark chamber was simulated by maintaining the glass slides inside Petri dishes wrapped up in aluminum foil.

Once slides were treated by IFA, they were examined under 400× magnification, using an epifluorescence microscope (BX41, Olympus®, Japan). The microscope was equipped with a fluorescein isothiocyanate optical filter set ($490nm$ and $520nm$ maximum excitation and emission wavelengths, respectively) and a camera (DP72, Olympus®, Japan).

6.2.3 Lens-free imaging and counting

For the sample preparation to be used on the lens-free microscope, the stock suspension was homogenized in a vortex mixer for $60s$, as shown in Figure 33. Then, a $30\mu L$ aliquot was deposited onto a chamber created for the LHM device. It consisted of a volume made by placing two coverslips glued on a glass slide and covered by another small glass slide, as illustrated in Figure 33.

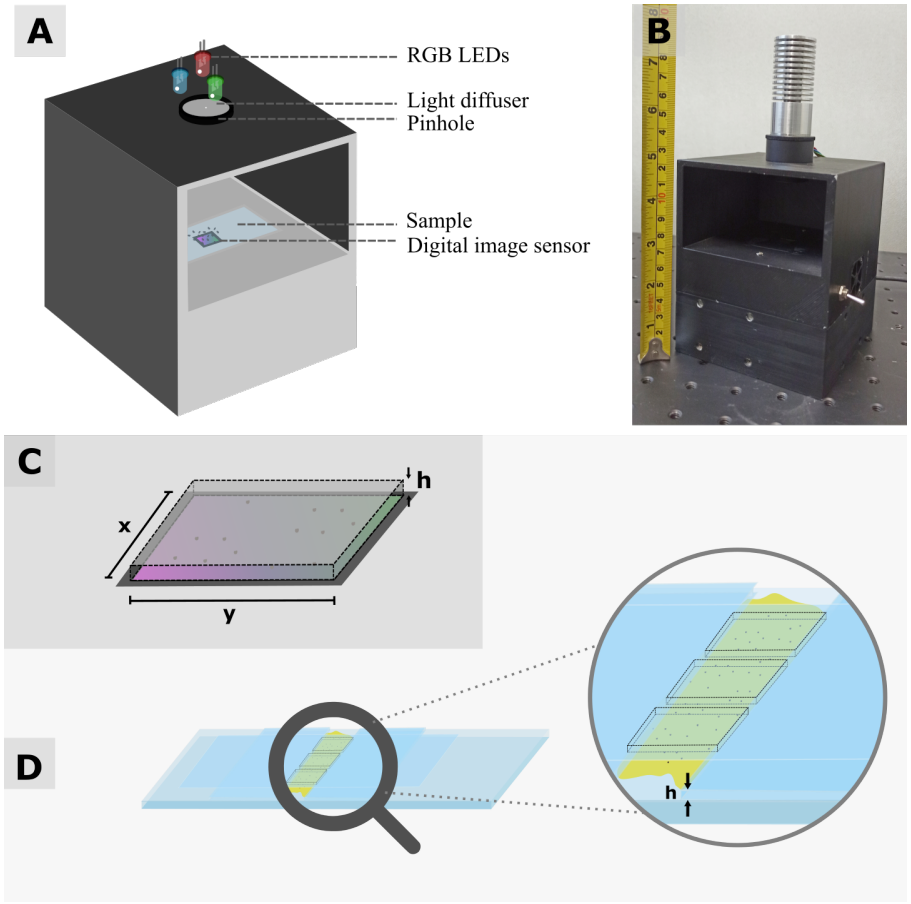

Figure 33 – Lens-free holographic microscope and its related imaging methods applied to observe the *Giardia* cysts solution. (a) Illustration and (b) photography of the lens-free holographic microscope used to record holograms with multi-wavelength partially coherent illumination. (c) Representation of the image volume. The lateral dimensions are equivalent to the sensor active area (x and y) and the height is equivalent to the thickness of the chamber where the sample was placed. (d) Representation of the homemade chamber used to perform the lens-free microscopic images. The height of this chamber is equal to the thickness of lateral coverslips glued on the glass slide and its width is defined by the distance of separation between these two coverslips. The elements represented with dotted outlines refer to image volumes.

Source: By the author.

Immediately after filling the chamber with the stock suspension, it was ready to be imaged by the lens-free microscope. The acquisition step was performed for three different

positions of the filled chamber, in order to capture most of the filled area, as shown in Figure 33. For each position, image capturing takes half a minute and provides three holograms of the same scene obtained with three different spectral ranges (red, green, and blue). Each sequence of three recorded holograms, obtained on the same position, was submitted to a digital image processing, which takes about $5min$ to provide the resulting image with the cysts counting. This final image has the same area as the original holograms (about $30mm^2$) and a micrometer resolution.

The digital image processing was performed in an AMD Ryzen 5 3500U PC, 2.10 GHz, 16.0 GB RAM by algorithms developed using the Python computational program language. It relies on two main parts: the first one is the hologram reconstruction, by phase recovery, and the second one refers to the detection and counting of *Giardia* cysts.

Input data for phase recovery considered three holograms recorded with different spectral ranges. However, before calculation, these were submitted to a preprocessing procedure: The first step consists in determining the focusing distance z_2. Even though z_2 is estimated experimentally, it is essential that this parameter is digitally adjusted to ensure that the elements of the reconstructed images are observed in their best condition.

To get z_2 adjusted, an autofocusing algorithm (Tamamitsu *et al.*, 2018) is implemented in each hologram. In order to reduce the computational cost of this processing step, just a cropped area of the original holograms is submitted to the autofocusing and posterior alignment. The adjusted parameter for distance and alignment is used on the original holograms to perform phase recovery calculations, which results in amplitude and phase images for each recorded scene recorded by the lens-free holographic microscope. (45)

Detection and counting of the *Giardia* cysts are performed by an algorithm developed with traditional imaging processing, using the amplitude images obtained by phase recovery. To achieve cyst detection, the first step is to enhance the image contrast using a gamma correction (with gamma factor = 0.5).(92) Then, an intensity threshold converts the grayscale image into a binary image, (92) resulting in a spatial mask with the cysts candidate areas in white and the background in black. Next, the candidate elements to be considered cysts are defined from the detection of the contours of the mask. The area calculation of that element follows the detection of each element. In this way, only elements with an area between 2 and 20-pixel size are detected as cysts. Finally, the algorithm counts all elements detected as cysts in the entire image.

Cyst concentration was defined by dividing the number of cysts detected on the entire field of view by the image volume, which is represented in Figure 33.

Note that the limit of detection (LOD) was calculated considering Equation 6.1. (106) The blank matrix was prepared by centrifuging the suspension at $3000rpm$ for

$15min$ (Centrifuge 5702, Eppendorf, Germany). $30\mu L$ from the supernatant was used to fill the chamber. Holograms were recorded using the same procedure as the other samples submitted to the LHM analysis. Then, these images were submitted to the same digital image processing described before, in order to apply the detection and counting algorithm on reconstructed amplitude images.

$$LOD = \mu + 3\sigma \tag{6.1}$$

in which μ is the mean value and σ is the standard deviation of the concentration calculated from blank matrix analysis.

6.2.4 Statistics

Enumeration of cysts from both microscopy methods was statistically compared to the information provided by the supplier by Student's t-test against a given mean (95% confidence interval). Shapiro-Wilk normality test ($\alpha = 0.05$) was used to check the probability distribution of the samples prior to applying statistical analysis. Inferential statistics were performed by PAST 3.2. (110)

6.3 Results and Discussion

6.3.1 Cyst detection and enumeration

Figure 34 displays the average concentration of cysts obtained by both standard fluorescence microscopy and the lens-free system. IFA results, initially acquired in terms of the number of cysts enumerated for a certain volume, were extrapolated to relative values of cysts/mL in order that these could be compared to the known average provided by the commercial supplier. The same extrapolation applies to microscopy readings by the lens-free device, in which, however, aliquots of the same volume were analyzed.

The mean cyst concentration determined by fluorescence microscopy and the known average for the purified suspension were significantly different ($p < 0.05$), suggesting that quantification by IFA underestimated the number of detected cysts. Underestimating cyst count in environmental samples, as in water or wastewater, may pose a threat to human health, given that this information is key to determining the need for contingency measures, e.g. extra disinfection steps, or even defining primary conceptualization of treatment utilities.

Peer research has shown that IFA kits may be misleading regarding the concentration of *Giardia* cysts due to low fluorescence signals for species such as *G. muris*,(111) causing similar issues in terms of underestimating risk. Our study has taken *G. duodenalis* as a target, a pathogen that led to proper staining when directly labeled by

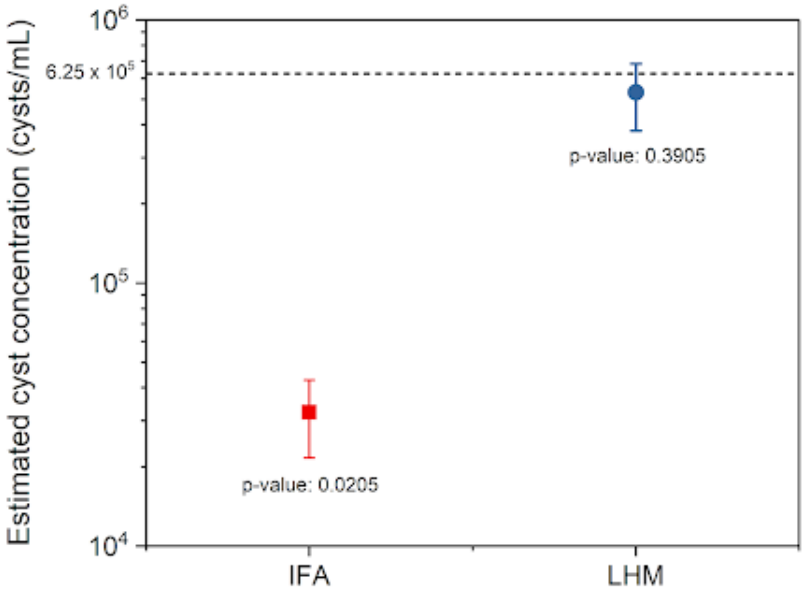

Figure 34 – Estimated cyst concentration from detection and counting performed by both methods (IFA and LHM). **Notes:** p-values refer to statistical analyses performed against a given mean of $625,000 cysts/mL$ (informed by the supplier and represented with the dashed line in this figure); IFA = immunofluorescence assay; LHM = lens-free holographic microscope.

Source: By the author.

USEPA-approved commercial kits. (111) Here, we may assign this underestimation to some points:

(i) operational/observer-dependant: it refers to the microscopists navigation throughout the well, neglecting certain areas. It is very likely to happen in immunofluorescence assays, which is why it is recommended at least two blind readings, in order to avoid bias and operational flaws. This is a matter that is not present in lens-free microscopy, as specific image fields are selected, covering most of the chamber.

(ii) protocol-inherent sample loss: cysts may be washed out of the microscopy well during the removal of dye excess, an issue that does not occur in the alternative technique, as the sample is contained within the glass slide chamber (shown in Figure 33) and no intermediate steps such as staining or washing are required.

(iii) quality of the sample: incorrect enumeration of clustered cysts, an issue that has been reported as impairing to immuno-based protocols, (102) at either the purification step (e.g. immunomagnetic separation) or IFA. Agglomeration of cysts (Figure 35) may hinder the attachment of antibodies to the cyst wall, as well as mislead the observer during

quantification. Clustered cysts are still a challenge for LHM detection, which can detect some cysts in the cluster, but hardly count them in its entirety. The presence of debris will also affect the efficiency of the lens-free microscope in detecting and quantifying cysts, but these will be filtered by intensity and area, as discussed further.

Though IFA refers to the recommended technique for quantifying cysts by both Methods 1623.1 and 1693, (99, 100) our results endorse the difficulties associated with it, particularly considering the high standard deviation found. This variability is often seen in studies that include the quality of recovery methods in terms of coefficient of variation, but they do not always consider the detection by IFA as something to be accounted for, as it tends to be a common step. A recent paper on detecting *Giardia* cysts in filter backwash water (105) compiled different recovery methods in complex matrices and showed that the coefficient of variation ranged from 3.2 to >100 in different works on commercial suspensions (spiked into water or residues), all of which were analyzed under fluorescence microscopy. Here, since no recovery was performed by either conventional or non-conventional methods, as we labeled already purified suspensions obtained from feces (higher cyst concentration than environmental matrices), fluorescence staining was carried out in ideal conditions, elucidating that even so there are issues in cyst enumeration.

The mean value concentration (Figure 34) obtained from lens-free microscopy is statistically similar ($p>0.05$) to the given value for the purified suspension. This result evidences the potential of LHM as a low-cost alternative to the detection and quantification of *Giardia* cysts.

LOD for the presented LHM analysis is $3613 cysts/mL$, which restricts the application of this presented method for purified suspensions with higher concentrations, with orders of magnitude that exceed the LOD value. Therefore, in this adequate condition, the advantages of LHM can be explored in order to simplify the procedures related to the quantification of *Giardia* cysts, from sample preparation to cyst counting. The present study intends to contribute to label-free, rapid, and low-cost alternatives which do not require trained specialists to operate. Among these alternatives, there is a friendly approach that uses images from a smartphone-based fluorescence microscope. (106) This technique presents a low LOD ($\sim 12 cysts/10mL$) and is a great alternative to evaluate large volumes ($10-20mL$) of water. However, it still relies on fluorescence, thus, expensive labeling remains a constraint. Another approach found in the literature describes an even simpler device to be coupled with a smartphone: Although it provides visual detection, it has a small field of view and does not offer a quantitative analysis yet. (112) Belini *et. al.* presented a robust instrumentation and algorithm to detect *Giardia* cysts in label-free sample solutions.(103) This in-situ microscopy also has a straightforward sample preparation method, besides low LOD and image resolution sufficient to enable a complete digital filters application on cysts detection analysis, even though it does not ensure that whole

the sample will be imaged either that there is no repeat count of the same cyst.

The use of LHM offers a label-free, quick, and straightforward sampling preparation since it requires just the cleaning and filling of the glass chamber with the homogeneous solution. The imaging method presented for the LHM approach analyzes a static sample in a wide field of view, which offers the capability of taking into account all the elements present in the image and not counting any element more than once. Furthermore, the automation applied to its image acquisition and digital image processing facilitates the use of this technique, besides providing reproducibility of measurements. The implementation of these digital algorithms also contributed to the reduction of the time interval required for it, performing holograms acquisition in $30s$, for each FOV, and digital image processing to detect and count cysts in less than $5min$.

All the cited advantages reinforce LHM as a potential tool for the detection and quantification of a purified suspended solution, even though this technique still requires future improvements to offer lower LOD and accurate digital detection.

6.3.2 General limitations and potentials

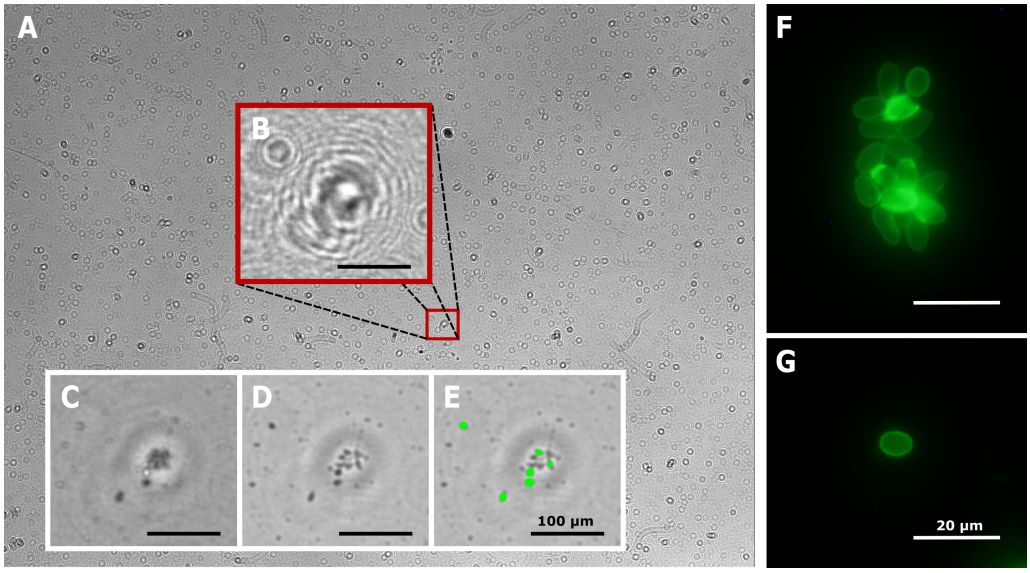

Figure 35 – Images of *Giardia* cysts obtained by lens-free holographic microscopy and fluorescence microscopy. (a) Raw hologram from LHM, whose area equals the system field of view. (b) Amplified image of a small area from hologram (a). (c) Phase image related to the hologram (b), showing clustered and isolated cysts. (d) Amplitude image related to the hologram (b). (e) Amplitude image after being submitted to the detection *Giardia* cyst algorithm; Green elements are detected as cysts. (f) Image of clustered cysts by fluorescence microscopy. (g) Image of an isolated cyst by fluorescence microscopy. White scale bars represent $20\mu m$ and black scale bars represent $100\mu m$.

Source: By the author.

Most limitations refer to the matrix to be imaged. Here we noticed difficulties associated with cyst agglomeration, as previously mentioned. In order to illustrate this issue, Figure 35.a displays a representative hologram from LHM and a region of interest with more details in the insert images. These cropped areas represent the hologram throughout the analysis: from the original hologram (Figure 35.b) to the post-processed hologram after the detection of cysts elements (Figure 35.e). Insert images show both clustered and isolated cysts, which are also presented on fluorescence images (Figure 35.f and 35.g, respectively). Particularly, images Figure 35.e and 35.f show a representation of some difficulties and capabilities inherent to each technique, concerning agglomerated cysts. As the clusters allow a 3D formation with cysts very close to each other, they pose challenges related to detection and counting by both standard IFA and LHM.

Table 3 summarizes the overall properties of both standard IFA and LHM on key elements to be considered in *Giardia* cyst visualization. The parameters point to the protocol simplification in time, infrastructure, and reproducibility of the method using LHM related to IFA. These may apply as either advantages or limitations depending on the goal and available resources, but definitely indicate the potential of the lens-free microscope for the detection of *Giardia* cysts, also encouraging future studies on more complex challenges.

6.4 Conclusions

Lens-free holographic microscopy is an effective alternative for detecting and quantifying *Giardia* cysts highly concentrated in purified samples. This study actually achieved significant similarity to the reference number of cysts, which was not obtained by standard immunofluorescence. Although the latter refers to the main protocol for water and disinfected wastewater samples, it also has issues associated with cost, time for the analysis, and user bias that are not present in LHM, elucidating the potential of this alternative technique.

Future directions for LHM targeting *Giardia* cysts should primarily target obtaining lower LOD. Also, it should focus on the efficiency of the lens-free microscope for the detection and quantification of cysts obtained from different environmental samples, treated by the variability of recovery methods available, as well as the USEPA standard protocols.

Table 3 – Overview of immunofluorescence assay (IFA) compared to lens-free holographic microscopy (LHM) considering sample preparation, enumeration, and additional features.

General features	**IFA**	**LHM**
Sample preparation	• Time consuming (sample drying and incubation) • Fluorescent reagents required • Additional laboratory equipment required (e.g. incubator) • Specialist operator required • Includes the possible loss of cysts in sample dying step • Must be carried out in the absence of direct light	• Rapid preparation ($< 1min$) • No reagents required • No additional equipment required • No specialist operator required • Does not include possible loss of cysts in sample dying step • No requirements about the room illumination
Visualization and quantification	• Benchtop and expensive microscope required • Observer-dependant detection and quantification, but easy to distinguish cysts against the background • Includes user bias, even being an antibody-based method • Does not allow visualization of internal structures • Time consuming ($\sim 30min/well$ depending on cyst concentration) • Difficulty of slide reuse	• Portable, automated, and low-cost equipment • Automated detection and quantification • No user bias • Visualization of internal structures is a topic for potential further research on confirmatory analysis • Rapid analysis ($\sim 5min/FOV$) • Chamber reuse
Cyst viability	• May be assessed by additional staining with fluorescent probes	• Possibility for future work
Limit of detection	• Considered inadequate by peer research (102, 113)	• $3613cysts/mL$

Source: By the author.

7 WOUND HEALING ASSAY

7.1 Introduction

Wound healing assay, also known as scratch assay, is a standard *in-vitro* method to assess and quantify collective cell migration, growth, and reproduction in two dimensions. In this assay, a cell-free area is created in a monolayer confluent cell culture. The gap area induces cells to migrate to cover all free space. Therefore, this movement is monitored over time. (114)

The scratch assay is a great tool to study collective cell migration, which is an important factor in wound repair, cancer invasion and metastasis, immune responses, angiogenesis, and embryonic morphogenesis. (115)

Usually, the temporal evolution of cell healing is recorded by microscopic images, from which the gap closure rate is obtained. (114) The image acquisition is typically performed by a transmitted-light microscope, which ranges in complexity from the traditional versions to more complete microscopes optimized for this application.

The use of more basic optical microscopes together with manual measurements based on visual analysis is tedious, imprecise, and prone to user bias. The addition of a digital camera contributes to the optimization of the analysis since it can offer information for quantitative analysis of cell dynamics. (116)

Measurements made in an environment without the necessary conditions for cell dynamics are the simplest option, as they depend only on a transmission light microscope, with which laboratories are usually equipped. Nonetheless, repeatedly taking the samples in and out of an incubator may compromise cell physiology because of variations in important parameters for cell maintenance. In those cases where the sample does not remain together with the microscope throughout the experiment, some visual references can be done using two scratched crosses in each well (117) or simply drawing reference lines on the bottom of the plate, (118) in order to image the same field of view over time.

Ideally, the microscope should be equipped with a digital camera, environmental chamber (that replicates a cell incubator by controlling temperature and humidity), motorized stage for multi-position acquisition, motorized focus, and image acquisition software. (114) Commercial equipment such as cell imaging readers can also be used for this purpose. (119) Further, as an alternative to this bulky and expensive equipment, lens-free holographic microscopes proved to be a great option to monitor the dynamic of cell growth, reproduction, and migration. (120) Lens-free holographic microscopy offers simple, lightweight, and low-cost instrumentation which provides low-coherence holographic images.(9) This type of instrumentation can be used inside a cell incubator to perform

cell monitoring, being an advantage for wound healing assays. In addition, the wide FOV of this imaging system optimizes the analysis of each sample. In this chapter, we present a portable and easy-to-use lens-free holographic microscope with automated acquisition and digital image processing for analysis of gap closure.

7.2 Material and Methods

7.2.1 Cell culture

An endothelial cell line, EA.hy926 (ATCC CRL-2922™) from the human umbilical vein, was used. It consists of an immortalized somatic cell hybrid resulting from a fusion of primary human umbilical vein cells (HUVEC) with a human lung carcinoma cells (A549) clone. EA.hy926 grows by adhesion in Dulbecco's Modified Eagle's Medium (DMEM), supplemented with 10% (v/v) of Fetal Bovine Serum (FBS).

For culture maintenance, cells were kept in a 5% CO_2 humidified atmosphere incubator at 37^oC, the culture medium was renewed every two to three days, and subculture was performed whenever cell confluence reached 80% to 90% of the flask area. The subculturing procedure was executed according to ATCC recommended protocol, using 0.25% (w/v) Trypsin - 0.53 mM EDTA with a seeding density of 2×10^3 to 3×10^3 viable $cells/cm^2$. In order to perform the experiments, cells were plated in glass bottom surface treated and sterile confocal dishes, with a $13mm$ diameter hole and a $34.30mm$ diameter adhesion area. Cells were inoculated at 2×10^6 cells per dish density to obtain a high confluence, which is crucial for wound healing monitoring.

7.2.2 Scratch-Wound Healing Assay

48h after inoculation, a wound was made in the confluent monocellular layer using a $200\mu L$ pipette tip, in a single flowing movement to obtain a clean wound edge, as shown in Figure 36. Right after scratching, the cellular medium was renewed and dishes were placed at a 37^oC and 5% CO_2 humidified atmosphere incubator, in which the lens-free holographic microscope was placed to monitor wound healing.

The prepared cell culture was placed on the lens-free microscope, in a 5% CO_2 humidified atmosphere incubator at 37^oC. To monitor the wound healing, images of the gap were taken every 30 min, for $25h$. We standardized the images to have the wound horizontally across the entire field of view of the microscope. During all the monitoring time, the lens-free microscope was connected to a portable computer, which was placed outside the cell incubator. A homemade software was developed in Python programming language to control the lens-free microscope and to automate this image acquisition process. More information about this software can be found in Chapter 4. The recorded images were saved with a sequential name, using leading zeros to differentiate units, tens,

and hundreds (i.e., 001, 010, and 100), to have a correct sequence to be interpreted by the image processing algorithm.

It is important to start the analysis after the cell dish was thermalized with the incubator atmosphere, in order to avoid noise effects on the image due to liquid condensation on the cover of the cell culture dish. Therefore, the presented analysis has beginning $1.5h$ after the wound was made.

7.2.3 Lens-free holographic microscope

A lens-free holographic microscope was built and used to record microscopic images directly from the cell culture. A photograph of the imaging system inside the cell incubator is shown in Figure 36.

The open part of the microscope hardware is positioned towards inside the cell incubator, to ensure that cell culture monitoring was not affected by opening the incubator door. Therefore, these experiments could be conducted without preventing the use of the incubator for other purposes.

Essentially, this lens-free microscope is composed of partially coherent illumination, a digital image sensor, and additional electronic components to perform the computational control of these devices. The partially coherent light comes from a LED ($518nm$) filtered by a spatial filter and a pinhole with $50\mu m$ of diameter (P50S, Thorlabs, USA). It illuminates the cell dish positioned right above the digital image sensor (CMOS $1.67\mu m$ pixel Aptina/Onsemi, USA). Thus, the sensor captures the hologram of the sample, which is the interference pattern of the light that passes through the sample without interacting with it and the light that interacts with the sample, by diffraction, for example.

The distance between the light source and the sample (z_1) is about $5cm$ and the distance between the sample and image sensor (z_2) is about $2mm$. With this configuration, we achieve a very large field of view (FOV), compared to the imaged areas of traditional optical microscopes. This FOV is almost equivalent to the image sensor active area used in this instrumentation ($\sim 30mm^2$).

This lens-free microscope is originally used to perform amplitude and phase information about the same scene of the observed sample, using RGB (red, blue, and green) LEDs. For these presented experiments, the raw images (holograms) registered with a single color (green) were enough to achieve the important information about wound healing, which allows further simplifying the instrumentation used for these experiments and saves the computational cost of post-processing the images.

A homemade software was developed in Python to automate the equipment. An algorithm controls the equipment and another algorithm uses the recorded holograms to perform the digital image processing to offer the gap area decay over time.

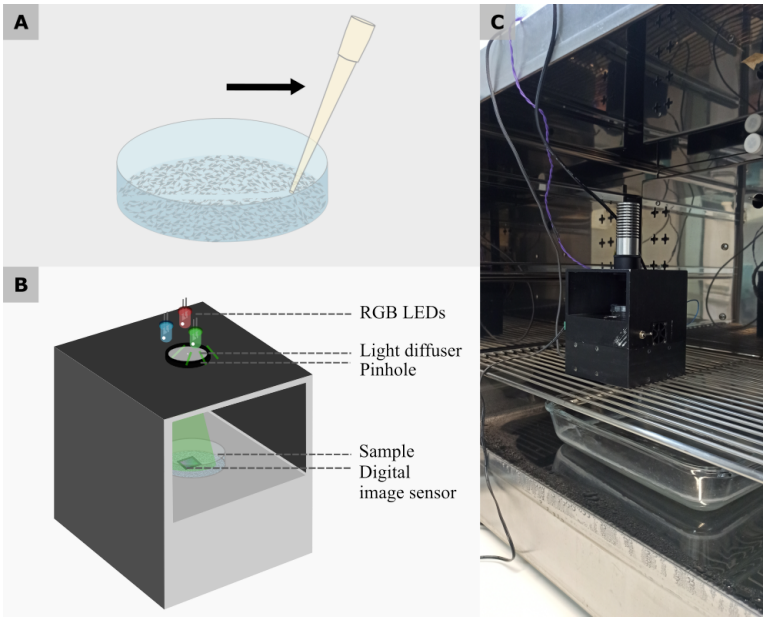

Figure 36 – Wound healing assay preparation and monitoring. (a) Schematic representation of the scratching on the cell culture using a pipette tip. (b) Schematic representation of the lens-free microscope used to image and monitor the wound on cell culture. (c) Photography of the lens-free imaging system inside the cell incubator to perform the imaging monitoring.

Source: By the author.

7.2.4 Traditional bright-field microscope

Nikon Eclipse-S, with a 4× objective lens, was also used to monitor gap closure. Two FOVs were recorded for each gap (n.1 and n.2), accordingly to the illustration in Figure 38.

This wound-healing assay was performed in triplicate. During the entire incubation time, one of the three samples (sample 2) was kept inside the lens-free (working) microscope. Thus, it was possible to verify whether there was any influence of this proposed image monitoring system on the gap closure.

7.2.5 Digital image processing

Digital image processing starts loading the sequence of holograms acquired every 30 min in the monitoring time interval. To lower computational cost, the original holograms (3664 × 2748 pixels) have their dimensions reduced (to 1000 × 750 pixels).

The next step is to increase the contrast and brightness of each redimensioned hologram. Then, to evidence the difference between cells and the gap area, a variance filter was applied. After this, the image is submitted to a threshold that converts it into a binary image, with the entire gap in black and most of the cell region in white. To find the gap edges, the algorithm finds all the contour elements on the image and lastly, applies

an area filter to avoid noise detection to be considered a gap area. Figure 37 exemplifies a hologram submitted to this described digital processing.

Unlike some automated digital image processing applied for wound healing assays, our algorithm considers more than one region per image to define the gap area. Therefore, in this present analysis, the gap area is defined as a sum of the detected gap regions in an image, which results in more accurate area values.

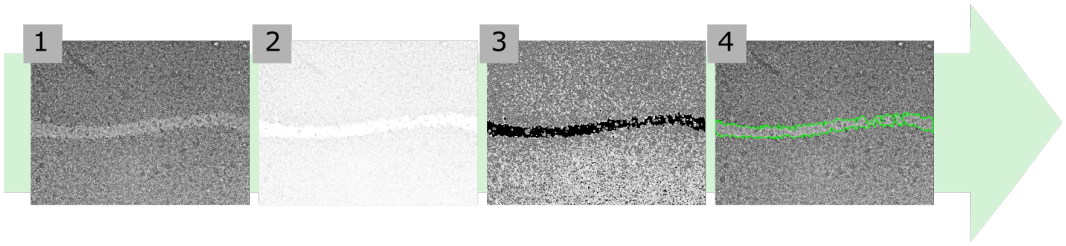

Figure 37 – A lens-free hologram, recorded on wound healing assay, throughout digital image processing for quantification of the gap area. The holograms are shown after being resized, in order of processing steps: (1) Monochrome hologram image (2) After contrast and brightness enhancement (3) With variance filter application (4) After detecting contours with an area greater than $200 pixels^2$.

Source: By the author.

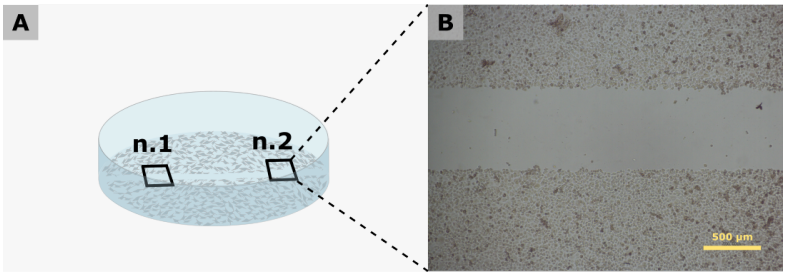

Figure 38 – Representation of the wound healing assay monitoring. (a) Schematic representation of the observed areas to perform the imaging monitoring using a traditional optical microscope. (b) The typical image (of an entire FOV) of the wound right after the scratching.

Source: By the author.

Considering the differences between holograms and bright-field images, they were submitted to different software in order to extract their information. The images recorded with the bright-field optical microscope were analyzed using an ImageJ plug-in (*Wound_healing_size_tool*), proposed by Suarez-Arnedo et.al. (118) Despite their particularities, these image processing define the gap area based on similar operations. Nonetheless, the image processing applied to the bright-field images considers only the larger area to be the gap area.

7.3 Results and Discussion

The lens-free holographic microscope was able to perform the wound healing assay without user supervision for more than $24h$. Throughout this time period, the cell culture remained in the same position in relation to the sensor that recorded the images. Thus, the system ensures the same region of the cell culture is being monitored.

The gaps have an extension more significant than the lateral dimension of the FOV and were horizontally aligned for the recordings of the holograms. Thus, the measurement of the free-cell area in the images is proportional to the mean width of this gap.

The results from the hologram analysis are represented in the bottom row of Figure 39, in which some holograms of a sequence of the image monitoring of the same scratch show the detected area with green edges. As this cell type does not reproduce in less than $24h$, only the migration contributes to the observed wound healing.

Figure 39 also shows, on the top row, the gap area detected on bright-field images. One can see the considerable difference between the field of view of both applied imaging techniques. Although the traditional optical microscope offers images with more details than the holograms, its observed area is smaller. The heterogeneity of the gap closure is observed for both areas presented in Figure 39, however, the reduced observed area could mitigate this lack of homogeneity. For this reason, in the case of bright-field images, it is convenient to analyze more than one FOV.

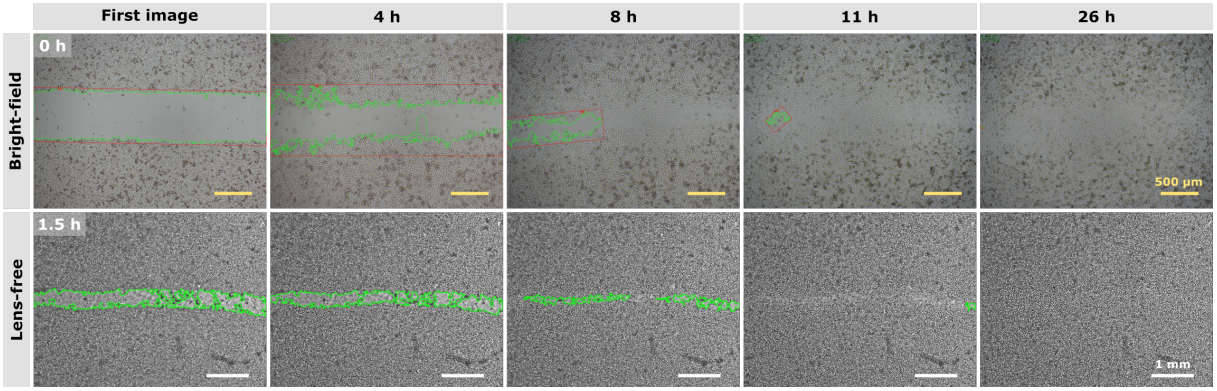

Figure 39 – Images from the bright-field microscope (top row) and lens-free microscope (bottom row) representing the wound healing analysis. The green lines denote the gap edges detected by the algorithms.

Source: By the author.

The images shown in a column on the right of Figure 40 present the detected gaps $1.5h$ after the scratching, when the hologram analysis started. As can be noted, the gaps had different initial widths, so the closure time was expected to be different for each one of the cell cultures. The graph presented in the same Figure 40 confirms

the difference of the gap area decay over time for each monitored sample. However, this time difference is not directly proportional to the size of the gap. Apparently, there is a tendency for closing to be slower as the gap width increases. However, a larger number of measurements are required to confirm this information and indicate the relationship between these two parameters (closing time and gap width).

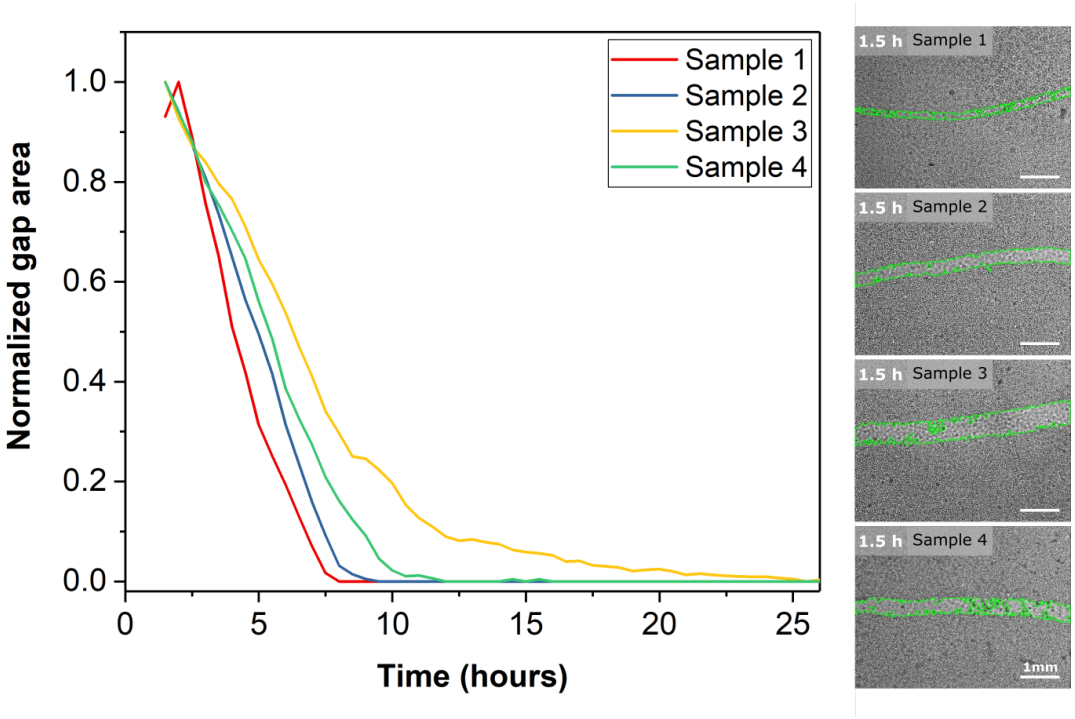

Figure 40 – Graph of wound healing assay performed by lens-free microscope and images recorded $1.5h$ after the scratching, when imaging recording begins.

Source: By the author.

Figure 41 presents the decay of the gap area in time for the bright-field microscopic images. This graph is composed of a reduced number of measurements, compared to the holographic analysis. It is due to the experimental limitation related to removing the cell culture from the incubator to perform the images. As well as in the hologram analysis, for bright-field microscopic images, gap closures also show differences, although all observed gaps were closed $26h$ after scratching. There are differences between the gap closure of the observed FOVs of the same scratch. These differences are about 10% for samples 2 and 3, but it is greater than 40% for the observed FOVs, in sample 1, for the first hours. The reduced number of measurements makes the analysis difficult and highlights possible area detection limitations associated with the algorithm used.

In the wound healing assay performed with image recordings out of the cell incubator, sample 2 does not demonstrate a different behavior when compared to the other ones of this same assay. This indicates that there is no influence of the lens-free microscope instrument on the wound-healing process for cell culture.

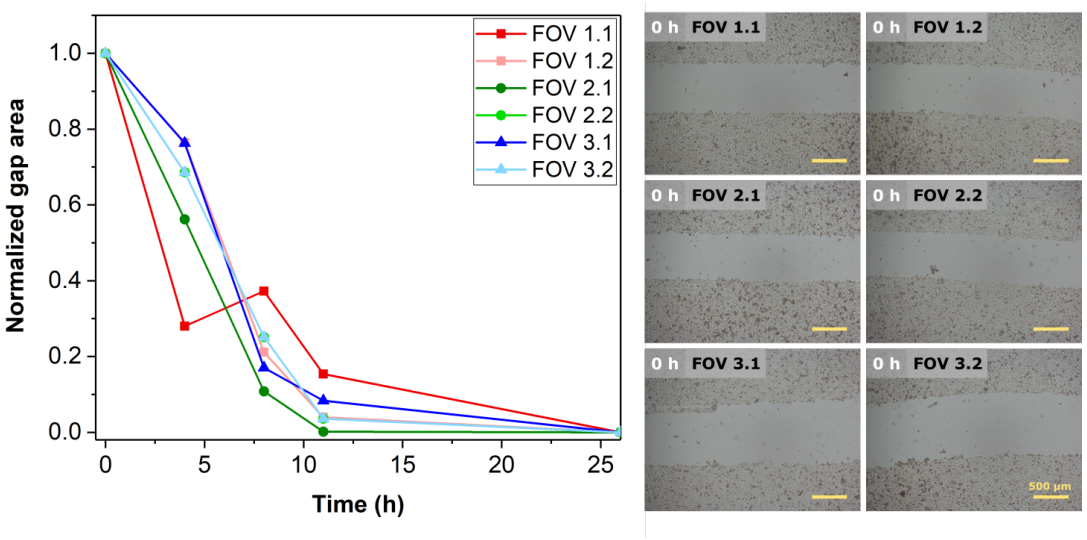

Figure 41 – Graph of wound healing assay performed using a bright-field microscope and their images recorded just after the scratching, when imaging recording begins. Images represent the two areas observed for each sample.

Source: By the author.

7.4 Conclusion

The presented lens-free holographic microscope is a portable, lightweight, and automated system that proved to be efficient to perform image monitoring on wound healing assays. This microscopy system can work inside a cell incubator and provides cell culture images with almost $30mm^2$ for hours or days without human supervision.

As the lens-free microscope is maintained inside the cell incubator, the cell culture does not need to be exposed to significant variations from its ideal developmental parameters. It reduces the influence of external factors on cell culture development, with is very important for scientific experiments.

This large FOV provides more information about the cell-free area over time when compared with the images from traditional optical microscopes, which contributes to optimized analysis. Additionally, the automation of image processing provides an objective and quantitative analysis of the samples.

In the future, this lens-free imaging system can be applied to assess the influence of treatments on cell cultures. Therefore, it can contribute to *in vitro* studies in order to optimize the parameters of each of these techniques applied to cells.

8 CONCLUSION

Two lens-free holographic microscopes were developed in this study, based on different phase recovery methods, one based on multiheight and the other based on multispectral method. The difference in methods implies distinction in the hardware that performs the acquisition of holograms and also in the software that digitally processes these images.

Both assemblies are portable (with dimensions less than $10cm \times 15cm \times 20cm$ for multiheight and $10cm \times 10cm \times 20cm$ for multispectral), lightweight (1095 g for multiheight and 606 g for the multispectral microscope), and have automated control. They also have a wide field of view, with an area $19\times$ larger than a typical microscope image that uses a $10\times$ objective lens.

These LHM systems were characterized using a resolution test target. The spatial resolution was experimentally defined for each lens-free microscope. The multiheigh and multispectral LHM system has a resolution of $\sim 4\mu m$ and $\sim 6\mu m$, respectively.

Open-source algorithms were developed in Python to perform the control of the microscope and to provide the phase recovery for each method. All the digital imaging processing was performed using traditional operations and took about $8min$ for multiheight and $4min$ for the multispectral phase recovery processing, including autofocusing and alignment for all the holograms.

Biological samples were used to test their performance, besides being used to explore the applications of the presented LHM systems. Both lens-free microscopes proved to be efficient for imaging monolayer cell cultures and other microorganisms such as yeast cells and microalgae in solution. The final version of the multispectral microscope also demonstrates its efficiency to work inside a cell incubator, providing imaging monitoring of the sample.

REFERENCES

1 BRADBURY, S. **The evolution of the microscope**. Berlin: Elsevier, 2014.

2 MURPHY, D. B. **Fundamentals of light microscopy and electronic imaging**. New York: Wiley-Liss, 2001. ISBN 047125391X.

3 ROY, M. *et al.* A review of recent progress in lens-free imaging and sensing. **Biosensors and Bioelectronics**, Elsevier, v. 88, p. 130–143, 2017.

4 MCLEOD, E.; OZCAN, A. Unconventional methods of imaging: computational microscopy and compact implementations. **Reports on Progress in Physics**, IOP Publishing, v. 79, n. 7, p. 076001, 2016.

5 GREENBAUM, A. *et al.* Imaging without lenses: achievements and remaining challenges of wide-field on-chip microscopy. **Nature Methods**, Nature Publishing Group, v. 9, n. 9, p. 889–895, 2012.

6 GARCIA-SUCERQUIA, J. Color digital lensless holographic microscopy: laser versus led illumination. **Applied Optics**, Optical Society of America, v. 55, n. 24, p. 6649–6655, 2016.

7 YU, Y.-W. *et al.* An edge-lit volume holographic optical element for an objective turret in a lensless digital holographic microscope. **Scientific Reports**, Nature Publishing Group, v. 10, n. 1, p. 1–9, 2020.

8 BOOMINATHAN, V. *et al.* Recent advances in lensless imaging. **Optica**, v. 9, n. 1, p. 1, 2022.

9 KIM, S. B. *et al.* Lens-Free imaging for biological applications. **Journal of Laboratory Automation**, v. 17, n. 1, p. 43–49, 2012.

10 OZCAN, A.; MCLEOD, E. Lensless imaging and sensing. **Annual Review of Biomedical Engineering**, Annual Reviews, v. 18, p. 77–102, 2016. DOI: 10.1146/annurev–bioeng–092515–010849.

11 KESAVAN, S. V. *et al.* High-throughput monitoring of major cell functions by means of lensfree video microscopy. **Scientific Reports**, Nature Publishing Group, v. 4, p. 5942, 2014.

12 WU, Y.-C. *et al.* Air quality monitoring using mobile microscopy and machine learning. **Light:** science & applications, Nature Publishing Group, v. 6, n. 9, p. e17046, 2017. DOI: 10.1038/lsa.2017.46.

13 TOBON-MAYA, H. *et al.* Open-source, cost-effective, portable, 3d-printed digital lensless holographic microscope. **Applied Optics**, OSA, v. 60, n. 4, p. A205–A214, 2021.

14 D'ALMEIDA, C. P. **Desenvolvimento e caracterização de um microscópio óptico holográfico sem lentes *in-line***. 2019. Dissertação (Mestrado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2018.

15 GABOR, D. A new microscopic principle. **Nature**, v. 161, n. 4098, p. 777–778, 1948.

16 GABOR, D. Holography , 1948-1971. **Science**, v. 177, n. 4046, p. 299–314, 1972.

17 SCHNARS, U.; JUPTNER, W. P. O. Digital recording and numerical reconstruction of holograms. **Measurement Science and Technology**, v. 13, n. 8, p. R85, 2002.

18 GESUALDI, M. R.; SOGA, D.; MURAMATSU, M. Real-time holographic interferometry using photorefractive sillenite crystals with phase-stepping technique. **Optics and Lasers in Engineering**, Elsevier, v. 44, n. 1, p. 56–67, 2006.

19 BRITO, I. V. *et al.* Photorefractive digital holographic microscopy applied in microstructures analysis. **Optics Communications**, Elsevier, v. 286, p. 103–110, 2013.

20 RICARDO, J. O. *et al.* Digital holography microscopy in 3D biologic samples analysis. **Journal of Physics:** conference series, v. 274, n. 1, p. 0–9, 2011.

21 GARCIA-SUCERQUIA, J. *et al.* Digital in-line holographic microscopy. **Applied Optics**, Optical Society of America, v. 45, n. 5, p. 836–850, 2006.

22 GABOR, D.; KOCK, W. E.; STROKE, G. W. Holography. **Science**, American Association for the Advancement of Science, v. 173, n. 3991, p. 11–23, 1971.

23 MUDANYALI, O. *et al.* Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications. **Lab on a Chip**, Royal Society of Chemistry, v. 10, n. 11, p. 1417–1428, 2010.

24 DELACROIX, R. *et al.* Cerebrospinal fluid lens-free microscopy: a new tool for the laboratory diagnosis of meningitis. **Scientific Reports**, Nature Publishing Group, v. 7, n. 1, p. 1–8, 2017.

25 MUDANYALI, O. *et al.* Detection of waterborne parasites using field-portable and cost-effective lensfree microscopy. **Lab on a Chip**, Royal Society of Chemistry, v. 10, n. 18, p. 2419–2423, 2010.

26 PUSHKARSKY, I. *et al.* Automated single-cell motility analysis on a chip using lensfree microscopy. **Scientific Reports**, Nature Publishing Group, v. 4, p. 4717, 2014.

27 GREENBAUM, A. *et al.* Wide-field computational imaging of pathology slides using lens-free on-chip microscopy. **Science Translational Medicine**, American Association for the Advancement of Science, v. 6, n. 267, p. 267ra175–267ra175, 2014.

28 ISIKMAN, S. O. *et al.* Lensfree on-chip microscopy and tomography for biomedical applications. **IEEE Journal of Selected Topics in Quantum Electronics**, IEEE, v. 18, n. 3, p. 1059–1072, 2011.

29 ROSTYKUS, M.; MOSER, C. Compact lensless off-axis transmission digital holographic microscope. **Optics Express**, Optical Society of America, v. 25, n. 14, p. 16652–16659, 2017.

30 SHIN, S.; YU, Y. Lensless reflection digital holographic microscope with a fresnel-bluestein transform. **Journal of the Korean Physical Society**, Springer, v. 74, n. 2, p. 98–101, 2019.

31 LEE, M.; YAGLIDERE, O.; OZCAN, A. Field-portable reflection and transmission microscopy based on lensless holography. **Biomedical Optics Express**, OSA, v. 2, n. 9, p. 2721–2730, 2011.

32 BERDEU, A. *et al.* Lens-free microscopy for 3d+ time acquisitions of 3d cell culture. **Scientific Reports**, Nature Publishing Group, v. 8, n. 1, p. 1–9, 2018.

33 WANG, M.; FENG, S.; WU, J. Multilayer pixel super-resolution lensless in-line holographic microscope with random sample movement. **Scientific Reports**, Nature Publishing Group, v. 7, n. 1, p. 1–8, 2017.

34 BISHARA, W. *et al.* Lensfree on-chip microscopy over a wide field-of-view using pixel super-resolution. **Optics Express**, Optical Society of America, v. 18, n. 11, p. 11181–11191, 2010.

35 LUO, W. *et al.* Pixel super-resolution using wavelength scanning. **Light:** science & applications, Nature Publishing Group, v. 5, n. 4, p. e16060–e16060, 2016.

36 ZHANG, J. *et al.* Adaptive pixel-super-resolved lensfree in-line digital holography for wide-field on-chip microscopy. **Scientific Reports**, Nature Publishing Group, v. 7, n. 1, p. 1–15, 2017.

37 HUANG, X. *et al.* Machine learning based single-frame super-resolution processing for lensless blood cell counting. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 11, p. 1836, 2016.

38 MANGAL, J. *et al.* Unsupervised organization of cervical cells using bright-field and single-shot digital holographic microscopy. **Journal of Biophotonics**, Wiley Online Library, v. 12, n. 8, p. e201800409, 2019.

39 RIVENSON, Y. *et al.* Phasestain: the digital staining of label-free quantitative phase microscopy images using deep learning. **Light:** science & applications, Nature Publishing Group, v. 8, n. 1, p. 1–11, 2019.

40 SINHA, A. *et al.* Lensless computational imaging through deep learning. **Optica**, Optical Society of America, v. 4, n. 9, p. 1117–1125, 2017.

41 RIVENSON, Y. *et al.* Phase recovery and holographic image reconstruction using deep learning in neural networks. **Light:** science & applications, Nature Publishing Group, v. 7, n. 2, p. 17141–17141, 2018.

42 GREENBAUM, A.; OZCAN, A. Maskless imaging of dense samples using pixel super-resolution based multi-height lensfree on- chip microscopy. **Optics Express**, v. 20, n. 3, p. 836–850, 2012.

43 STAHL, R. *et al.* Lens-free digital in-line holographic imaging for wide field-of-view, high-resolution and real-time monitoring of complex microscopic objects. **Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues XII**, v. 8947, p. 89471F, March 2014.

44 GREENBAUM, A.; OZCAN, A. Maskless imaging of dense samples using pixel super-resolution based multi-height lensfree on- chip microscopy. **Optics Express**, v. 20, n. 3, p. 836–850, 2012.

45 ALLIER, C. *et al.* Imaging of dense cell cultures by multiwavelength lens-free video microscopy. **Cytometry Part A**, Wiley Online Library, v. 91, n. 5, p. 433–442, 2017.

46 BIAN, Y. *et al.* Portable multi-spectral lens-less microscope with wavelength-self-calibrating imaging sensor. **Optics and Lasers in Engineering**, Elsevier, v. 111, p. 25–33, 2018. DOI: 10.1016/j.optlaseng.2018.07.009.

47 GöRöCS, Z. *et al.* A deep learning-enabled portable imaging flow cytometer for cost-effective, high-throughput, and label-free analysis of natural water samples. **Light:** science & applications, v. 7, n. 1, 2018.

48 GREENBAUM, A.; SIKORA, U.; OZCAN, A. Field-portable wide-field microscopy of dense samples using multi-height pixel super-resolution based lensfree imaging. **Lab on a Chip**, v. 12, n. 7, p. 1242, 2012.

49 SOBIERANSKI, A. C. *et al.* Portable lensless wide-field microscopy imaging platform based on digital inline holography and multi-frame pixel super-resolution. **Light:** science & applications, v. 4, n. 10, p. e346, 2015.

50 UO, Z. H. L. *et al.* Pixel super-resolution for lens-free holographic microscopy using deep learning neural networks. **Optics Express**, v. 27, n. 10, p. 13581–13595, 2019.

51 WU, X. *et al.* Pixel-super-resolution lensfree microscopy based on multiple-wavelength scanning. **Proceedings of SPIE**, v. 1157114, 2020. DOI: 10.1117/12.2580523.

52 VAGHASHIYA, R. *et al.* Machine learning based lens-free shadow imaging technique for field-portable cytometry. **Biosensors**, v. 12, n. 3, p. 1–15, 2022.

53 FEIZI, A. *et al.* Yeast viability and concentration analysis using lens-free computational microscopy and machine learning. **Proceedings of SPIE**, v. 10055, p. 1005508, 2017. DOI: 10.1117/12.2252731.

54 HIGHAM, D. J.; HIGHAM, N. J. **MATLAB guide**. 3rd ed. Philadelphia: Siam, 2016.

55 GOODMAN, J. W. **Introduction to Fourier optics**. 3rd ed. Englewood, CO: Roberts & Co. Publishers, 2005.

56 KIM, M. K. Principles and techniques of digital holographic microscopy microscopy. **SPIE Reviews**, v. 1, n. 1, 2010.

57 TAMAMITSU, M. *et al.* A robust holographic autofocusing criterion based on edge sparsity: comparison of gini index and tamura coefficient for holographic autofocusing based on the edge sparsity of the complex optical wavefront. **Proceedings of SPIE**, v. 10503, p. 105030J, 2018. DOI: 10.1117/12.2291179.

58 ZHANG, J. *et al.* Resolution analysis in a lens-free on-chip digital holographic microscope. **IEEE Transactions on Computational Imaging**, IEEE, v. 6, p. 697–710, 2020. DOI:10.1109/TCI.2020.2964247.

59 GÖRING, L. *et al.* Lensless digital holographic microscope using in-line configuration and laser diode illumination. **Proceedings of SPIE**, v. 10127, p. 101270V, 2017. DOI: 10.1117/12.2250927.

60 FEIZI, A. *et al.* Rapid, portable and cost-effective yeast cell viability and concentration analysis using lensfree on-chip microscopy and machine learning. **Lab on a Chip**, Royal Society of Chemistry, v. 16, n. 22, p. 4350–4358, 2016.

61 KABIR, M. A. *et al.* Automated detection of apoptotic versus nonapoptotic cell death using label-free computational microscopy. **Journal of Biophotonics**, v. 15, n. 4, p. 1–10, 2022.

62 EDMUND Optics. **Introduction to modulation transfer function**. 2011. Available from: https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-modulation-transfer-function/. Accessible at: 23 Jan. 2022.

63 GERCHBERG, R. W.; SAXTON, W. O. Practical algorithm for determination of phase from image and diffraction plane pictures. **Optik**, 35, n. 2, p. 237–&, 1972.

64 ANACONDA Software Distribution. Anaconda Inc. 2020. Anaconda Software Distribution. Available from: https://docs.anaconda.com/. Accessible at: 03 Mar. 2023.

65 HERTZ, L.; JUURLINK, B. H. J.; SZUCHET, S. Cell cultures. *In:* LAJTHA, A. (ed.). **Handbook of neurochemistry**. 2nd ed. New York: Plenum Press, 1985. v. 8. p. 603–661.

66 ATCC. **Primary dermal fibroblast normal; human, neonatal (HDFn) (ATCC® PCS-201-010™)**. 2021. Available from: https://www.atcc.org/products/pcs--201--010. Accessible at: 22 Feb. 2023.

67 THERMOFISHER Scientific. **Cell morphology**. 2023. Available from: https://www.thermofisher.com/br/en/home/references/gibco--cell--culture--basics/cell--morphology.html. Accessible at: 22 Feb. 2023.

68 ATCC. **SCC-25 (ATCC® CRL-1628™)**. 2021. Available from: https://www.atcc.org/products/crl--1628. Accessible at: 22 Feb. 2023.

69 BAIN, B. J. Diagnosis from the blood smear. **New England Journal of Medicine**, Mass Medical Soc, v. 353, n. 5, p. 498–507, 2005.

70 FACULTY of Biological Sciences at Leeds. Histology guide. **Red blood cells**. 2003. Available from: https://www.histology.leeds.ac.uk/blood/blood_rbc.php. Accessible at: 23 Feb. 2023.

71 FACULTY of Biological Sciences at Leeds. Histology guide. **White blood cells**. 2003. Available from: https://www.histology.leeds.ac.uk/blood/blood_wbc.php. Accessible at: 23 Feb. 2023.

72 FANG, Y. *et al.* An on-chip instrument for white blood cells classification based on a lens-less shadow imaging technique. **PLoS ONE**, v. 12, n. 3, p. 1–14, 2017.

73 ISIKMAN, S. *et al.* Lensfree cell holography on a chip: from holographic cell signatures to microscopic reconstruction. *In:* IEEE LEOS ANNUAL MEETING CONFERENCE, 2009, Belek-Antalya. **Proceedings** [...] Belek-Antalya: IEEE, 2009. p. 404–405 .

74 ROY, M. *et al.* A simple and low-cost device performing blood cell counting based on lens-free shadow imaging technique. **Sensors and Actuators, B:** chemical, Elsevier B.V., v. 201, p. 321–328, 2014. DOI: 10.1016/j.snb.2014.05.011.

75 FANG, Y.; YU, N.; LIU, Y. Classification of white blood cells by convolution neural network in lens-free imaging system. INTERNATIONAL CONGRESS ON IMAGE AND SIGNAL PROCESSING, BIOMEDICAL ENGINEERING AND INFORMATICS, CISP-BMEI, 11., 2018, Beijing. **Proceedings** [. . .] Beijing: IEEE, IEEE, 2019. p. 1–5.

76 SEO, S. *et al.* High-throughput lens-free blood analysis on a chip. **Analytical Chemistry**, v. 23, n. 1, p. 1–7, 2010.

77 CORRÊA, T. Q. **Técnicas ópticas para o controle microbiológico de sangue.** 2017. Tese (Doutorado em Biotecnologia) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, São Carlos, 2017.

78 RUANE, J.; SONNINO, A.; AGOSTINI, A. Bioenergy and the potential contribution of agricultural biotechnologies in developing countries. **Biomass and Bioenergy**, Elsevier Ltd, v. 34, n. 10, p. 1427–1439, 2010. DOI: 10.1016/j.biombioe.2010.04.011.

79 YAP, J. K. *et al.* A review manuscript submitted to Chemosphere advancement of green technologies: a comprehensive review on the potential application of microalgae biomass. **Chemosphere**, Elsevier Ltd, v. 281, p. 130886, May 2021. DOI: 10.1016/j.chemosphere.2021.130886.

80 ZHUANG, D. *et al.* Application progress of bioactive compounds in microalgae on pharmaceutical and cosmetics. **Chemosphere**, Elsevier Ltd, v. 291, n. P2, p. 132932, 2022. DOI: 10.1016/j.chemosphere.2021.132932.

81 WANG, C. *et al.* The intrinsic characteristics of microalgae biofilm and their potential applications in pollutants removal — a review. **Algal Research**, Elsevier B.V., v. 68, p. 102849, July 2022. DOI: 10.1016/j.algal.2022.102849.

82 KHANRA, A. *et al.* Green bioprocessing and applications of microalgae-derived biopolymers as a renewable feedstock: circular bioeconomy approach. **Environmental Technology and Innovation**, Elsevier B.V., v. 28, p. 102872, 2022. DOI: 10.1016/j.eti.2022.102872.

83 SEO, D. *et al.* High-throughput and real-time microalgae monitoring platform using lens-free shadow imaging system (LSIS). *In:* IEEE SENSORS, 2015, Busan. **Proceedings** [. . .] Busan: IEEE, 2015. p. 3-5.

84 SHIN, S. *et al.* Field-portable seawater toxicity monitoring platform using lens-free shadow imaging technology. **Water Research**, Elsevier Ltd, v. 230, p. 119585, December 2023. DOI: 10.1016/j.watres.2023.119585.

85 WANG, J. *et al.* Detection of viability of micro-algae cells by optofluidic hologram pattern. **Biomicrofluidics**, v. 12, n. 2, p. 1–10, 2018. DOI: 10.1063/1.5021179.

86 GORELOVA, O. A. *et al.* Similarity and diversity of the Desmodesmus spp. microalgae isolated from associations with White Sea invertebrates. **Protoplasma**, v. 252, n. 2, p. 489–503, 2015.

87 LI, X.; DREHER, T. W.; LI, R. An overview of diversity, occurrence, genetics and toxin production of bloom-forming Dolichospermum (Anabaena) species. **Harmful Algae**, Elsevier B.V., v. 54, p. 54–68, October 2016. DOI: 10.1016/j.hal.2015.10.015.

88 LEPPÄNEN, J. J. An overview of cladoceran studies conducted in mine water impacted lakes. **International Aquatic Research**, v. 10, n. 3, p. 207–221, 2018.

89 SUNAYAMA, K.; MIYAKAWA, H.; HAYASAKI, Y. Size measurement of daphnia pulex using low-coherence gabor digital holography. **Optical Review**, Springer, v. 26, n. 6, p. 693–698, 2019.

90 ABREU, M. J. *et al.* The use of PCR-RFLP to genetically distinguish the morphologically close species: Ceriodaphnia dubia Richard, 1894 and ceriodaphnia silvestrii Daday, 1902 (Crustacea Cladocera). **Brazilian Journal of Biology**, v. 70, n. 1, p. 121–124, 2010.

91 ZHANG, M. *et al.* Improvement of cell counting method for neubauer counting chamber. **Journal of Clinical Laboratory Analysis**, Wiley Online Library, v. 34, n. 1, p. e23024, 2020.

92 GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X.

93 SAITO, T. L. *et al.* SCMD: Saccharomyces cerevisiae Morphological Database. **Nucleic Acids Research**, Oxford University Press., v. 32, p. 319–322, 2004.

94 COELHO, C. H. *et al.* Giardiasis as a neglected disease in Brazil: systematic review of 20 years of publications. **PLoS Neglected Tropical Diseases**, Public Library of Science San Francisco, CA USA, v. 11, n. 10, p. e0006005, 2017.

95 HOTEZ, P. J. Neglected infections of poverty in the united states of america. **PLoS Neglected Tropical Diseases**, Public Library of Science San Francisco, USA, v. 2, n. 6, p. e256, 2008.

96 HOTEZ, P. J.; GURWITH, M. Europe's neglected infections of poverty. **International Journal of Infectious Diseases**, Elsevier, v. 15, n. 9, p. e611–e619, 2011.

97 CACCIÒ, S. M. *et al.* Giardia cysts in wastewater treatment plants in italy. **Applied and Environmental Microbiology**, Am Soc Microbiol, v. 69, n. 6, p. 3393–3398, 2003.

98 ORTEGA, Y. R.; ADAM, R. D. Giardia: overview and update. **Clinical Infectious Diseases**, JSTOR, v. 25, n. 3, p. 545–549, 1997.

99 USEPA. **Method 1623.1:** cryptosporidium and giardia in water by filtration/IMS/FA. 2012. Available from: http://water.epa.gov/scitech/drinkingwater/labcert/upload/epa816r12001.pdf. Accessible at: 23 Jan. 2021.

100 USEPA. **Method 1693:** cryptosporidium and giardia in disinfected wastewater by concentration/IMS/IFA. 2014. Available from: https://www.epa.gov/sites/default/files/2015-08/documents/method_1693_2014.pdf. Accessible at: 23 Jan. 2021.

101 LUKA, G. *et al.* Comprehensive review of conventional and state-of-the-art detection methods of cryptosporidium. **Journal of Hazardous Materials**, Elsevier, v. 421, p. 126714, 2022. DOI: 10.1016/j.jhazmat.2021.126714.

102 SILVA, K. J. S.; SABOGAL-PAZ, L. P. Giardia spp. cysts and cryptosporidium spp. oocysts in drinking water treatment residues: comparison of recovery methods for quantity assessment. **Environmental Technology**, Taylor & Francis, v. 42, n. 20, p. 3144–3153, 2021.

103 BELINI, V. L. *et al.* Label-free detection and enumeration of giardia cysts in agitated suspensions using in situ microscopy. **Journal of Microbiological Methods**, Elsevier, v. 199, n. 10, p. 106509, 2022.

104 FRADETTE, M.-S.; CULLEY, A. I.; CHARETTE, S. J. Detection of cryptosporidium spp. and giardia spp. in environmental water samples: a journey into the past and new perspectives. **Microorganisms**, Multidisciplinary Digital Publishing Institute, v. 10, n. 6, p. 1175, 2022.

105 OGURA, A. P.; SABOGAL-PAZ, L. P. Direct centrifugation for detecting giardia spp. cysts in filter backwash water. **Journal of Water, Sanitation and Hygiene for Development**, v. 12, n. 6, p. 475–484, 2022.

106 KOYDEMIR, H. C. *et al.* Rapid imaging, detection and quantification of Giardia lamblia cysts using mobile-phone based fluorescent microscopy and machine learning. **Lab on a Chip**, v. 15, n. 5, p. 1284–1293, 2015.

107 SHRESTHA, R. *et al.* A smartphone microscopic method for simultaneous detection of (Oo)cysts of cryptosporidium and giardia. **PLoS Neglected Tropical Diseases**, v. 14, n. 9, p. 1–19, 2020. DOI: 10.1371/journal.pntd.0008560.

108 WU, Y.; OZCAN, A. Lensless digital holographic microscopy and its applications in biomedicine and environmental monitoring. **Methods**, Elsevier, v. 136, p. 4–16, 2018. DOI: 10.1016/j.ymeth.2017.08.013.

109 SCHOLZ, G. *et al.* Continuous live-cell culture imaging and single-cell tracking by computational lensfree led microscopy. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 5, p. 1234, 2019.

110 HAMMER, Ø. *et al.* Past: paleontological statistics software package for education and data analysis. **Palaeontologia Electronica**, College Station, v. 4, n. 1, p. 9, 2001.

111 ALDERISIO, K. A. *et al.* Differences in staining intensities affect reported occurrences and concentrations of giardia spp. in surface drinking water sources. **Journal of Applied Microbiology**, Wiley Online Library, v. 123, n. 6, p. 1607–1613, 2017.

112 SHRESTHA, R. *et al.* A smartphone microscopic method for simultaneous detection of (oo) cysts of cryptosporidium and giardia. **PLoS Neglected Tropical Diseases**, Public Library of Science San Francisco, CA USA, v. 14, n. 9, p. e0008560, 2020.

113 ELSAFI, S. H. *et al.* Comparison of microscopy, rapid immunoassay, and molecular techniques for the detection of giardia lamblia and cryptosporidium parvum. **Parasitology Research**, Springer, v. 112, n. 4, p. 1641–1646, 2013.

114 JONKMAN, J. E. *et al.* An introduction to the wound healing assay using live-cell microscopy. **Cell Adhesion and Migration**, v. 8, n. 5, p. 440–451, 2014.

115 GRADA, A. *et al.* Research techniques made simple: analysis of collective cell migration using the wound healing assay. **Journal of Investigative Dermatology**, v. 137, n. 2, p. e11–e16, 2017. DOI: 10.1016/j.jid.2016.11.020.

116 CORTESI, M. *et al.* AIM: A computational tool for the automatic quantification of scratch wound healing assays. **Applied Sciences (Switzerland)**, v. 7, n. 12, p. 1237, 2017.

117 GEBÄCK, T. *et al.* TScratch: a novel and simple software tool for automated analysis of monolayer wound healing assays. **BioTechniques**, v. 46, n. 4, p. 265–274, 2009.

118 SUAREZ-ARNEDO, A. *et al.* An image J plugin for the high throughput image analysis of in vitro scratch wound healing assays. **PLoS ONE**, v. 15, n. 7, p. 1–14, 2020.

119 GARCIA-FOSSA, F.; GAAL, V.; JESUS, M. B. PyScratch: an ease of use tool for analysis of scratch assays. **Computer Methods and Programs in Biomedicine**, Elsevier B.V., v. 193, 2020.

120 MOMEY, F. *et al.* Dynamics of cell and tissue growth acquired by means of extended field of view lensfree microscopy. **Biomedical Optics Express**, v. 7, n. 2, p. 512, 2016.

ANNEX

ANNEX A – PHASE RECOVERY PYTHON CODE FOR MULTIHEIGHT METHOD

```
'''
*** Multiheight Python code - alignment and phase reconstruction ***
        ~ for entire images (all Field of View) ~
'''

# ------------- Libraries  ---------------------- #
import cv2                          # Library for image processing
import numpy as np                  # Library for math
import os                           # Library for organizing archives
from matplotlib import pyplot as plt # Library for display imgs
import time                         # Library for time counting

# ------------- Functions ---------------------- #
# Function to rescale an image
def rescale(img, scale_factor):
    width = img.shape[1]
    ImgScale = scale_factor/width
    newX, newY = img.shape[1]*ImgScale, img.shape[0]*ImgScale
    image = cv2.resize(img,(int(newX),int(newY)))
    Height,Width = image.shape
    return image, ImgScale

# Global variables that are used on the click_and_crop function
refPt = [] # vector to save the selected reference coordinates
cropping = False

# click_and_crop shows an image and saves the reference coordinates of
the user-drawn rectangle
def click_and_crop(event, x, y, flags, param):
    global refPt, cropping

    # event is a user action
    if event == cv2.EVENT_LBUTTONDOWN: # reference initial coordinate
set on the user click with
        refPt = [(x,y)]                # left mouse button
        cropping = True

    elif event == cv2.EVENT_LBUTTONUP:  # reference final coordinate
set on the user click with
        refPt.append((x,y))             # left mouse button
        cropping = False
        # draw a retangle with the reference coordinates
        cv2.rectangle(image, refPt[0], refPt[1], (0, 255, 0), 2)
        cv2.imshow("image", image)

# load_images opens the selected folder and returns a list with all
images that it has inside
def load_images(folder):
    I = []

    for filename in os.listdir(folder):
```

```
        img = cv2.imread(os.path.join(folder,filename), 0)
        if img is not None:
            I.append(img)
    return I

# crop images with selected reference coordinates
def crop(image):
    clone = image.copy()
    return
clone[int(refPt[0][1]/Scale):int(refPt[1][1]/Scale),int(refPt[0][0]/Sca
le):int(refPt[1][0]/Scale)]#função que corta nos pontos de referência

def Interactive_propagation(val):
    global img_crop, z_value, lambda_value
    lambda_value = cv2.getTrackbarPos('Lambda', window_name)
    z_value = cv2.getTrackbarPos('10z', window_name)
    z_value = z_value/10

    Uimg = U(img_crop,z_value*1000,lambda_value*0.001)

    img_show = rescale_intensity(np.abs(Uimg))
    cv2.imshow(window_name, img_show)

def alignment(im1,im2,im2_fov):
    sz = im2_fov.shape
    # Define the motion model
    warp_mode = cv2.MOTION_TRANSLATION

    # Define 2x3 or 3x3 matrices and initialize the matrix to identity
    if warp_mode == cv2.MOTION_HOMOGRAPHY :
        warp_matrix = np.eye(3, 3, dtype=np.float32)
    else :
        warp_matrix = np.eye(2, 3, dtype=np.float32)

    # Specify the number of iterations.
    number_of_iterations = 5000;

    # Specify the threshold of the increment
    # in the correlation coefficient between two iterations
    termination_eps = 1e-10;

    # Define termination criteria
    criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT,
number_of_iterations,  termination_eps)

    # Run the ECC algorithm. The results are stored in warp_matrix.
    (cc, warp_matrix) = cv2.findTransformECC(im1,im2,warp_matrix,
warp_mode, criteria, inputMask=None, gaussFiltSize=1)

    if warp_mode == cv2.MOTION_HOMOGRAPHY :
```

```
        im_aligned = cv2.warpPerspective (im2_fov, warp_matrix,
(sz[1],sz[0]), flags=cv2.INTER_LINEAR + cv2.WARP_INVERSE_MAP)
    else:
        im_aligned = cv2.warpAffine(im2_fov, warp_matrix,
(sz[1],sz[0]), flags=cv2.INTER_LINEAR + cv2.WARP_INVERSE_MAP)

    return im_aligned

def imgradient(img):
    grad_x = cv2.Sobel(img, cv2.CV_8UC1, 1,0)
    grad_y = cv2.Sobel(img, cv2.CV_8UC1, 0,1)
    abs_grad_x = cv2.convertScaleAbs(grad_x)
    abs_grad_y = cv2.convertScaleAbs(grad_y)
    grad = cv2.addWeighted(abs_grad_x, 0.5, abs_grad_y, 0.5, 0)
    return grad

def rescale_intensity(img):
    min_val,max_val=img.min(),img.max()
    img = 255.0*(img - min_val)/(max_val - min_val)
    img = img.astype(np.uint8)
    return img


# ------------- Parameters ---------------------- #
# -------- units: micrometers ------------------
dx = 1.67; dy = dx          # pixel size
refractive_index = 1        # refractive index - medium between sample
to the sensor
num_iterations = 10         # number of iterations
deltaz = 50                 # distance between captured holograms

# ------------- Loading images ---------------------- #
Original_imgs =
load_images(r"C:\Users\Camila\Downloads\leveduras_azul_metileno_exp900"
)


# --------- Selecting cropping area ---------------- #
img = Original_imgs[0]

# Rescaling image to display
image, Scale = rescale(img, 1000)
clone = image.copy()
cv2.namedWindow("image")
cv2.setMouseCallback("image",click_and_crop)

while True:
    cv2.imshow("image",image)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("r"):
```

```
        image = clone.copy()

    elif key == ord("c"):
        break

if len(refPt)==2:
    crop_img = clone[refPt[0][1]:refPt[1][1],refPt[0][0]:refPt[1][0]]
    newimg, imgScale = rescale(crop_img, 1000)


# --------- Cropping Original images ---------------- #
Cropped_imgs = []
for img in Original_imgs:
    Cropped_imgs.append(crop(img))


# -------- Propagation - Cropped images ------------- #

# image to be used as reference for functions of propagation
img_crop = Cropped_imgs[0]
[Nx,Ny] = img_crop.shape

# Frequence domain matrix
f = np.linspace(0,0+Nx, Nx,endpoint=False)
fxv = np.fft.ifftshift((f-np.floor(Nx/2+1))*(1/(Nx*dx)))
f = np.linspace(0,0+Ny, Ny,endpoint=False)
fyv = np.fft.ifftshift((f-np.floor(Ny/2+1))*(1/(Ny*dx)))
fx, fy = np.meshgrid(fxv, fyv, sparse=False, indexing='ij')

# Transfer function
def TF(z,Lambda):
    tf =
np.exp((1j*refractive_index*2*np.pi*z*np.sqrt((1/Lambda)**2-fx**2-fy**2
)))
    return tf

def U(Ui,z,Lambda):
    Uift = np.fft.fft2(Ui)
    u = np.fft.ifft2(np.multiply(Uift,TF(z,Lambda)))
    return u

# ---------------- Autofocus ------------------------ #
def autofocus(img, z2_min, z2_max, n_z2, Lambda):
    hU = np.sqrt(img)
    deltaz= (z2_max - z2_min)/n_z2
    z2_vector = np.arange(z2_min,z2_max,deltaz)
    tot_img = z2_vector.size
    U0 = []
    TC_list = []
    for i in range(tot_img):
        U0.append(np.abs(U(hU, z2_vector[i], Lambda)))
```

```
        grad = imgradient(U0[i])
        mean = np.mean(grad)
        std = grad.std()
        TC = np.sqrt(std/mean)
        TC_list.append(TC)

    TC_max = max(TC_list)
    a = 0
    for i in range(len(TC_list)):
        if TC_list[i] == TC_max:
            a = i
    return z2_vector[a]

# ------ trackbar to define z2_vector and lambda --------- #
window_name = 'Propagation Controls for Amplitude Image'
# Create window
cv2.namedWindow(window_name)
# Create a Trackbar to choose variable values
cv2.createTrackbar('Lambda', window_name, 632, 700,
Interactive_propagation)
cv2.createTrackbar('10z', window_name , 8, 28, Interactive_propagation)
# Call the function to initialize
Interactive_propagation(0)
# Wait until the user finishes the program
cv2.waitKey()

start = time.time() ########################################

# Redifine parameters
Lambda = lambda_value/1000         # selected lambda in micrometers
z_value = z_value*1000             # selected z2 in micrometers
dz2 = 0.01                         # step between z2 values
n_z2 = 50                          # number of divisions to be
considered between z2_min and z2_max
z2_min = z_value - (n_z2//2)*dz2   # mininum distance to be used as
input of autofocus function
z2_max = z_value + (n_z2//2)*dz2   # maximum distance to be used as
input of autofocus function


z2 = []
for a in range(len(Cropped_imgs)):
    z2.append(autofocus(Cropped_imgs[a], z2_min, z2_max, n_z2, Lambda))

OnFocus_imgs = []
for a in range(len(Cropped_imgs)):

OnFocus_imgs.append(np.uint8(np.absolute(U(Cropped_imgs[a],z2[a],Lambda
))))

# -------- Alignment - On Focus images ------------- #
```

```
Aligned_imgs = []
Aligned_imgs.append(Original_imgs[0])
for i in range(len(OnFocus_imgs)):
    if i != 0:

Aligned_imgs.append(alignment(OnFocus_imgs[0],OnFocus_imgs[i],Original_
imgs[i]))
print("Alignment done \n")


# ****** Ploting On Focus (Cropped) images **********
fig = plt.figure(figsize=(10, 7))

fig.add_subplot(1, 3, 1)            # (rows, columns, position)
plt.imshow(img_crop,cmap='gray')
plt.axis('off')
plt.title("Hologram")

fig.add_subplot(1, 3, 2)            # (rows, columns, position)
plt.imshow(np.absolute(U(Cropped_imgs[0],z2[0], Lambda)),cmap='gray')
plt.axis('off')
plt.title("Amplitude - On focus")

fig.add_subplot(1, 3, 3)            # (rows, columns, position)
plt.imshow(np.angle(U(Cropped_imgs[0],z2[0], Lambda)),cmap='gray')
plt.axis('off')
plt.title("Phase - On focus")

plt.show()


# -------- Multiheight Phase Recorvering ------------ #
# --------- Cropping and Aligned holograms ------------- #
I = []
for i in range(len(Aligned_imgs)):
    I.append(Aligned_imgs[i])

# -------- Propagation - Cropped images ------------- #

# image to be used as reference for functions of propagation
img = Original_imgs[0]
[Nx,Ny] = img.shape

# Frequence domain matrix
f = np.linspace(0,0+Nx, Nx,endpoint=False)
fxv = np.fft.ifftshift((f-np.floor(Nx/2+1))*(1/(Nx*dx)))
f = np.linspace(0,0+Ny, Ny,endpoint=False)
fyv = np.fft.ifftshift((f-np.floor(Ny/2+1))*(1/(Ny*dx)))
fx, fy = np.meshgrid(fxv, fyv, sparse=False, indexing='ij')
```

```
hU = []
U0 = []

for a in range(len(I)):
    hU.append(np.sqrt(I[a]))
    U0.append(U(hU[a],z2[a], Lambda))


print("Starting multiheight imaging processing")
print("..........................................")

for i in range(num_iterations):
    print("iteration: ", i+1)

    for a in range(0,len(I)-1):
        U0[a+1] = U(U0[a], -deltaz, Lambda)
        U0[a+1] = np.multiply(hU[a+1],np.exp(1j*np.angle(U0[a+1])))

    for a in range(len(I)-1,0,-1):
        U0[a-1] = U(U0[a], deltaz, Lambda)
        U0[a-1] = np.multiply(hU[a-1],np.exp(1j*np.angle(U0[a-1])))


Hologram_img = Original_imgs[0]
Amplitude_img = np.abs(U(U0[0],z2[0], Lambda))
Phase_img = np.angle(U(U0[0],z2[0], Lambda))

# ****** Ploting Final Cropped images **********
fig = plt.figure(figsize=(10, 7))

fig.add_subplot(1, 3, 1)            # (rows, columns, position)
plt.imshow(Hologram_img,cmap='gray')
plt.axis('off')
plt.title("Hologram")

fig.add_subplot(1, 3, 2)            # (rows, columns, position)
plt.imshow(Amplitude_img,cmap='gray')
plt.axis('off')
plt.title("Amplitude")

fig.add_subplot(1, 3, 3)            # (rows, columns, position)
plt.imshow(Phase_img,cmap='gray')
plt.axis('off')
plt.title("Phase")

plt.show()


# Running time
end = time.time()   ###################################
```

```
print("\n""Running time: ", round((end-start)//60),"min e ",
round((end-start)%60), "s" )

cv2.imwrite('Amplitude_final.png',rescale_intensity(Amplitude_img))
cv2.imwrite("Phase_final.png",rescale_intensity(Phase_img))
cv2.imwrite("OnFocus.png",rescale_intensity(np.abs(U(Original_imgs[0],z
2[0], Lambda))))
```

ANNEX B – PYTHON CODE TO CONTROL DEVICES FOR MULTIHEIGHT METHOD

```
"""
*** Python code to control the multiheight lens-free microscope ***
    Control of the camera (Arducam MT9J001) and actuator
"""

# ------------- Libraries  --------------------- #
import sys
import time
import os
import threading

import pylibftdi
import pyAPT

import cv2
import numpy as np
import signal
import json
from ImageConvert import *
import ArducamSDK

global
cfg,handle,running,Width,Heigth,saved_flag,color_mode,save_raw,home_pos
ition,serial,delta_z, rtn_val, con, translation, z, folder, exposure


# ---------- CONTROL PARAMETERS --------------- #
z = 0                 ## Auxiliary variable to control the image
acquisition
translation = True    ## True if Translation is off and can be used
running     = True    ## True if the program is running
saved_flag  = True    ## True to save the current image
cfg = {}
handle = {}

# ---------- ADJUSTABLE PARAMETERS --------------- #
save_raw      = False       ## If the user wants the images to be
registered with .raw format
serial        = '83854962'  ## Actuator serial number
home_position = 7.3500      ## Initial position of the camera
n_pictures    = 6           ## Number of images to be registered
delta_z       = 0.05        ## Moving step for the camera [mm]
exposure      = 2           ## Initial value for exposure values

# Current address
addr = os.getcwd()

# ------------- Functions --------------------- #
# Set the configurations of the camera board
def configBoard(fileNodes):
    global handle
```

```
    for i in range(0,len(fileNodes)):
        fileNode = fileNodes[i]
        buffs = []
        command = fileNode[0]
        value = fileNode[1]
        index = fileNode[2]
        buffsize = fileNode[3]
        for j in range(0,len(fileNode[4])):
            buffs.append(int(fileNode[4][j],16))

ArducamSDK.Py_ArduCam_setboardConfig(handle,int(command,16),int(value,1
6),int(index,16),int(buffsize,16),buffs)

pass

# Write on configuration camera registers
def writeSensorRegs(fileNodes):
    global handle
    for i in range(0,len(fileNodes)):
        fileNode = fileNodes[i]
        if fileNode[0] == "DELAY":
            time.sleep(float(fileNode[1])/1000)
            continue
        regAddr = int(fileNode[0],16)
        val = int(fileNode[1],16)

        if regAddr != 12306: ## Adiciona a exposição da câmera que
desejamos
            ArducamSDK.Py_ArduCam_writeSensorReg(handle,regAddr,val)
        else:

ArducamSDK.Py_ArduCam_writeSensorReg(handle,regAddr,exposure)

pass

# Read the JSON file and set the parameters
def camera_initFromFile(fialeName):
    global cfg,handle,Width,Height,color_mode,save_raw
    #load config file
    config = json.load(open(fialeName,"r"))

    camera_parameter = config["camera_parameter"]
    Width = int(camera_parameter["SIZE"][0])
    Height = int(camera_parameter["SIZE"][1])

    BitWidth = camera_parameter["BIT_WIDTH"]
    ByteLength = 1
    if BitWidth > 8 and BitWidth <= 16:
        ByteLength = 2
        save_raw = True
    FmtMode = int(camera_parameter["FORMAT"][0])
```

```
    color_mode = (int)(camera_parameter["FORMAT"][1])

    I2CMode = camera_parameter["I2C_MODE"]
    I2cAddr = int(camera_parameter["I2C_ADDR"],16)
    TransLvl = int(camera_parameter["TRANS_LVL"])
    cfg = {"u32CameraType":0x4D091031,
            "u32Width":Width,"u32Height":Height,
            "usbType":0,
            "u8PixelBytes":ByteLength,
            "u16Vid":0,
            "u32Size":0,
            "u8PixelBits":BitWidth,
            "u32I2cAddr":I2cAddr,
            "emI2cMode":I2CMode,
            "emImageFmtMode":FmtMode,
            "u32TransLvl":TransLvl }

    # ArducamSDK.
    #ret,handle,rtn_cfg = ArducamSDK.Py_ArduCam_open(cfg,0)
    ret,handle,rtn_cfg = ArducamSDK.Py_ArduCam_autoopen(cfg)
    if ret == 0:

        #ArducamSDK.Py_ArduCam_writeReg_8_8(handle,0x46,3,0x00)
        usb_version = rtn_cfg['usbType']
        #print("USB VERSION:",usb_version)
        #config board param
        configBoard(config["board_parameter"])

        if usb_version == ArducamSDK.USB_1 or usb_version ==
ArducamSDK.USB_2:
            configBoard(config["board_parameter_dev2"])
        if usb_version == ArducamSDK.USB_3:
            configBoard(config["board_parameter_dev3_inf3"])
        if usb_version == ArducamSDK.USB_3_2:
            configBoard(config["board_parameter_dev3_inf2"])

        writeSensorRegs(config["register_parameter"])

        if usb_version == ArducamSDK.USB_3:
            writeSensorRegs(config["register_parameter_dev3_inf3"])
        if usb_version == ArducamSDK.USB_3_2:
            writeSensorRegs(config["register_parameter_dev3_inf2"])

        rtn_val,datas =
ArducamSDK.Py_ArduCam_readUserData(handle,0x400-16, 16)
        '''print("Serial:
%d%d%d%d-%d%d%d%d-%d%d%d%d"%(datas[0],datas[1],datas[2],datas[3],

datas[4],datas[5],datas[6],datas[7],

datas[8],datas[9],datas[10],datas[11]))
```

```
        '''
        return True
    else:
        print("open fail,rtn_val = ",hex(ret))
        return False

pass


# Perform image capture and add FIFO)
def captureImage_thread():
    global handle,running

    rtn_val = ArducamSDK.Py_ArduCam_beginCaptureImage(handle)
    if rtn_val != 0:
        print("Error beginning capture, rtn_val = ",hex(rtn_val))
        running = False
        return
    #else:
        #print("Capture began, rtn_val = ",rtn_val)

    while running:
        #print "capture"
        rtn_val = ArducamSDK.Py_ArduCam_captureImage(handle)
        if rtn_val > 255:
            print("Error capture image, rtn_val = ",hex(rtn_val))
            if rtn_val == ArducamSDK.USB_CAMERA_USB_TASK_ERROR:
                break
        time.sleep(0.005)

    running = False
    ArducamSDK.Py_ArduCam_endCaptureImage(handle)


# Get the image from FIFO, perform the conversion, and show the image
def readImage_thread():
    global
handle,running,Width,Height,saved_flag,cfg,color_mode,save_raw, folder,
addr, exposure, con, z
    global
COLOR_BayerGB2BGR,COLOR_BayerRG2BGR,COLOR_BayerGR2BGR,COLOR_BayerBG2BGR
    time0 = time.time()
    time1 = time.time()
    data = {}

    cv2.namedWindow("Lens Free",1)

    while running:
        display_time = time.time()
        if ArducamSDK.Py_ArduCam_availableImage(handle) > 0:
```

```
            rtn_val,data,rtn_cfg =
ArducamSDK.Py_ArduCam_readImage(handle)
            datasize = rtn_cfg['u32Size']

            if rtn_val != 0:
                print("read data fail!")
                continue

            if datasize == 0:
                continue

            image = convert_image(data,rtn_cfg,color_mode)

            time1 = time.time()
            if time1 - time0 >= 1:
                #print("%s %d %s\n"%("fps:",count,"/s"))
                count = 0
                time0 = time1

            ## salva a imagem na folder
            if not saved_flag:
                if not
os.path.exists(addr+"\\images\\"+folder+"_exp"+str(exposure)):

os.makedirs(addr+"\\images\\"+folder+"_exp"+str(exposure))


if(cv2.imwrite(addr+"\\images\\"+folder+"_exp"+str(exposure)+"\\image%d
.jpg"%z,image)):
                    print("Imagem gravada com sucesso!\n")
                else:
                    print("Erro ao gravar a imagem!\n")

                if save_raw:
                    with
open(addr+"\\images\\"+folder+"_exp"+str(exposure)+"\\image%d.raw"%z,
'wb') as f:
                        f.write(data)


                saved_flag = True


            image = cv2.resize(image,(640,480),interpolation =
cv2.INTER_LINEAR)

            cv2.imshow("Lens Free",image)
            cv2.waitKey(10)
            ArducamSDK.Py_ArduCam_del(handle)
            #print("-----------------------display time:",(time.time()
- display_time))
```

```
        else:
            time.sleep(0.001);



# Performs the translation and controls the image acquisition
def translation_thread():

    global saved_flag, delta_z, running, z, translation, home_position,
con

    while running:
        if translation == False:

            if(z<(n_pictures)):
                saved_flag = False

                if (saved_flag):
                    if(z<n_pictures-1):
                        con.move(delta_z)
                        time.sleep(3)
                        print('Step %d - Position: %.2f'
%(z+1,con.position()))
                    #time.sleep(3)
                    #saved_flag = False
                    z+=1
            else:
                con.goto(home_position)
                translation = True
                print("End of the acquisition process.\n")
                z=0

def menu():

    print("\n======= LENS FREE MICROSCOPY =======\n")
    print("Command options:\n")
    print("  I - Change intensity values")
    print("  S - Start multiheight acquisition")
    print("  C - Close program\n")

def parameters():

    print('\n. Number of images: %s' %n_pictures)
    print('. Exposure time: %s' %exposure)
    print('. Delta_z: %.2f\n' %delta_z)

# ------------ Main function -------------------- #
def main():

    global running, translation, con, folder, exposure
```

```
    flag_erro = False

    print("\n======= LENS FREE MICROSCOPY =======\n")

    # Adress of JSON
    config_file_name = addr+"\\Json\\MT9J001_10MP.json"

    if not os.path.exists(config_file_name):
        print("File JSON not found.”)
        exit()


    # Starts the actuator
    print("Starting components. Please, wait ... \n")
    print('--- Actuator ---')
    try:
        con = pyAPT.MTS50(serial_number = serial)

        print('\nStatus: OK!')
        con.goto(home_position)

        print('. Initial position: %.2f' %(con.position()))

    except:
        print('>>>>> Status: Error')
        print(' Actuator %s not found'%serial)
        print(' Please, check if the actuator is plugged in or if it is
connected to the computer.\n')
        flag_erro = True

    # Starts the camera
    print('\n--- Camera ---\n')
    if camera_initFromFile(config_file_name):
        ArducamSDK.Py_ArduCam_setMode(handle,
ArducamSDK.CONTINUOUS_MODE)
        print('> Status: Ok!\n')
    else:
        print('>>>>> Status: Error')
        print(" Please, check if the camera driver has been installed
correctly or if the camera is connected.\n")
        flag_erro = True

    if (flag_erro):
        exit()


    # Create the threads
    ci = threading.Thread(target = captureImage_thread)
    rt = threading.Thread(target = readImage_thread)
    tt = threading.Thread(target = translation_thread)
```

```
    # Start the treads associate with the capture and read the images
    ci.start()
    rt.start()

    while running:

        if translation:
            time.sleep(1)
            menu()

            input_kb = str(sys.stdin.readline()).strip("\n")

            if input_kb == 'i' or input_kb == 'I': ## Exposure values
changing
                print("Enter the new exposure value:")
                exposure = sys.stdin.readline().strip("\n")

ArducamSDK.Py_ArduCam_writeSensorReg(handle,12306,int(exposure))

            elif input_kb == 'c' or input_kb == 'C': ## Close program
running = False

            elif (input_kb == 's' or input_kb == "S"): ## Starts the
image acquisition
                folder = input("Insert folder name:”)
                print("\nAcquisition started with the following
parameters:")
                parameters()
                time.sleep(2)
                print('. Step %d - Position: %.2f' %(z,con.position()))
                translation = False

                if not tt.is_alive():
                    tt.start()

    ci.join()
    rt.join()

    if tt.is_alive():
        tt.join()

    print("Closing components. Please, wait ...\n")

    rtn_val = ArducamSDK.Py_ArduCam_close(handle)

    con.home()

    if(con.position() == 0):
        print('> Actuator closed successfully!')
    else:
        print('>>>>> ERROR to close the actuator.')
```

```
    if rtn_val == 0:
        print('> Camera closed successfully!\n')
    else:
        print('>>>>> ERROR to close camera.\n')


if __name__ == '__main__':
    sys.exit(main())
```

ANNEX C – PHASE RECOVERY PYTHON CODE FOR MULTISPECTRAL METHOD

```
'''
*** Multispectral Python code - alignment and phase reconstruction ***
          ~ for entire images (all Field of View) ~
'''

# ------------- Libraries  ---------------------- #
import cv2                           # Library for image processing
import numpy as np                   # Library for math
import os                            # Library for organizing archives
from matplotlib import pyplot as plt # Library for display images
import time                          # Library for time counting

# ------------- Functions ---------------------- #
# Function to rescale an image
def rescale(img, scale_factor):
    width = img.shape[1]
    ImgScale = scale_factor/width
    newX, newY = img.shape[1]*ImgScale, img.shape[0]*ImgScale
    image = cv2.resize(img,(int(newX),int(newY)))
    Height,Width = image.shape
    return image, ImgScale

# Global variables that are used on click_and_crop function
refPt = [] # vector to save the selected reference coordinates
cropping = False

# click_and_crop shows an image and saves the reference coordinates of
the user-drawn rectangle
def click_and_crop(event, x, y, flags, param):
    global refPt, cropping

    # event is a user action
    if event == cv2.EVENT_LBUTTONDOWN: # reference initial coordinate
set on the user click with
        refPt = [(x,y)]                # left mouse button
        cropping = True

    elif event == cv2.EVENT_LBUTTONUP:  # reference final coordinate
set on the user click with
        refPt.append((x,y))             # left mouse button
        cropping = False
        # draw a rectangle with the reference coordinates
        cv2.rectangle(image, refPt[0], refPt[1], (0, 255, 0), 2)
        cv2.imshow("image", image)

# load_images open the selected folder and return a list with all
images that it has inside
def load_images(folder):
    I = []

    for filename in os.listdir(folder):
```

```
        img = cv2.imread(os.path.join(folder,filename), 0)
        if img is not None:
            I.append(img)
    return I

# To crop images with selected reference coordinates
def crop(image):
    clone = image.copy()
    return
clone[int(refPt[0][1]/Scale):int(refPt[1][1]/Scale),int(refPt[0][0]/Sca
le):int(refPt[1][0]/Scale)]#função que corta nos pontos de referência

def Interactive_propagation(val):
    global img_crop, z_value, Lambdas
    z_value = (cv2.getTrackbarPos('10*z', window_name))/10

    Uimg = U(img_crop,z_value*1000, Lambdas[0])

    img_show = rescale_intensity(np.absolute(Uimg))
    cv2.imshow(window_name, img_show)

def alignment(im1,im2,im2_fov):
    sz = im2_fov.shape
    # Define the motion model
    warp_mode = cv2.MOTION_TRANSLATION

    # Define 2x3 or 3x3 matrices and initialize the matrix to identity
    if warp_mode == cv2.MOTION_HOMOGRAPHY :
        warp_matrix = np.eye(3, 3, dtype=np.float32)
    else :
        warp_matrix = np.eye(2, 3, dtype=np.float32)

    # Specify the number of iterations.
    number_of_iterations = 5000;

    # Specify the threshold of the increment
    # in the correlation coefficient between two iterations
    termination_eps = 1e-10;

    # Define termination criteria
    criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT,
number_of_iterations,  termination_eps)

    # Run the ECC algorithm. The results are stored in warp_matrix.
    (cc, warp_matrix) = cv2.findTransformECC(im1,im2,warp_matrix,
warp_mode, criteria, inputMask=None, gaussFiltSize=1)

    if warp_mode == cv2.MOTION_HOMOGRAPHY :
        im_aligned = cv2.warpPerspective (im2_fov, warp_matrix,
(sz[1],sz[0]), flags=cv2.INTER_LINEAR + cv2.WARP_INVERSE_MAP)
    else:
```

```
        im_aligned = cv2.warpAffine(im2_fov, warp_matrix,
(sz[1],sz[0]), flags=cv2.INTER_LINEAR + cv2.WARP_INVERSE_MAP)

    return im_aligned

def imgradient(img):
    grad_x = cv2.Sobel(img, cv2.CV_8UC1, 1,0)
    grad_y = cv2.Sobel(img, cv2.CV_8UC1, 0,1)
    abs_grad_x = cv2.convertScaleAbs(grad_x)
    abs_grad_y = cv2.convertScaleAbs(grad_y)
    grad = cv2.addWeighted(abs_grad_x, 0.5, abs_grad_y, 0.5, 0)
    return grad

def rescale_intensity(img):
    min_val,max_val=img.min(),img.max()
    img = 255.0*(img - min_val)/(max_val - min_val)
    img = img.astype(np.uint8)
    return img


# ------------- Parameters ---------------------- #
# -------- units: micrometers ------------------
dx = 1.67; dy = dx          # pixel size
refractive_index = 1        # refractive index - medium between sample
to the sensor
R = 0.632                   # red lambda
G = 0.518                   # green lambda
B = 0.462                   # blue lambda
Lambdas = [R,G,B]           # wavelengths used to capture the holograms

# ------------- Loading images ---------------------- #
Original_imgs =
load_images(r"C:\Users\Camila\Downloads\USAF_difusor_plastico_int1.00")


# --------- Selecting cropping area ---------------- #
img = Original_imgs[0]

# Rescaling image to display
image, Scale = rescale(img, 1000)
clone = image.copy()
cv2.namedWindow("image")
cv2.setMouseCallback("image",click_and_crop)

while True:
    cv2.imshow("image",image)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("r"):
        image = clone.copy()
```

```
    elif key == ord("c"):
        break

if len(refPt)==2:
    crop_img = clone[refPt[0][1]:refPt[1][1],refPt[0][0]:refPt[1][0]]
    newimg, imgScale = rescale(crop_img, 1000)
    #cv2.imshow("crop", newimg)
    #cv2.waitKey(0)
    #cv2.destroyAllWindows()


# --------- Cropping Original images ---------------- #
Cropped_imgs = []
for img in Original_imgs:
    Cropped_imgs.append(crop(img))


# -------- Propagation - Cropped images ------------ #

# image to be used as reference for functions of propagation
img_crop = Cropped_imgs[0]
[Nx,Ny] = img_crop.shape

# Frequence domain matrix
f = np.linspace(0,0+Nx, Nx,endpoint=False)
fxv = np.fft.ifftshift((f-np.floor(Nx/2+1))*(1/(Nx*dx)))
f = np.linspace(0,0+Ny, Ny,endpoint=False)
fyv = np.fft.ifftshift((f-np.floor(Ny/2+1))*(1/(Ny*dx)))
fx, fy = np.meshgrid(fxv, fyv, sparse=False, indexing='ij')

# Transfer function
def TF(z,Lambda):
    tf =
np.exp((1j*refractive_index*2*np.pi*z*np.sqrt((1/Lambda)**2-fx**2-fy**2
)))
    return tf

def U(Ui,z,Lambda):
    Uift = np.fft.fft2(Ui)
    u = np.fft.ifft2(np.multiply(Uift,TF(z,Lambda)))
    return u

# ---------------- Autofocus ------------------------ #
def autofocus(img, z2_min, z2_max, n_z2, Lambda):
    hU = np.sqrt(img)
    deltaz= (z2_max - z2_min)/n_z2
    z2_vector = np.arange(z2_min,z2_max,deltaz)
    tot_img = z2_vector.size
    U0 = []
    TC_list = []
    for i in range(tot_img):
```

```
        U0.append(np.abs(U(hU, z2_vector[i], Lambda)))
        grad = imgradient(U0[i])
        mean = np.mean(grad)
        std = grad.std()
        TC = np.sqrt(std/mean)
        TC_list.append(TC)

    TC_max = max(TC_list)
    a = 0
    for i in range(len(TC_list)):
        if TC_list[i] == TC_max:
            a = i
    return z2_vector[a]


# ----------- trackbar to define z2_vector ----------------- #
window_name = 'Propagation Controls'
# Create window
cv2.namedWindow(window_name)
# Create a Trackbar to choose variable values
cv2.createTrackbar('10*z', window_name , 8, 28,
Interactive_propagation)
# Call the function to initialize
Interactive_propagation(0)
# Wait until user finishes program
cv2.waitKey()

start = time.time() ###################################

z_value = z_value*1000             # selected z2 in micrometers
dz2 = 0.01                         # step bettween z2 values
n_z2 = 50                          # number of divisions to be consider
between z2_min and z2_max
z2_min = z_value - (n_z2//2)*dz2   # mininum distance to be used as
input of autofocus function
z2_max = z_value + (n_z2//2)*dz2   # maximum distance to be used as
input of autofocus function

z2 = []
for a in range(len(Cropped_imgs)):
    z2.append(autofocus(Cropped_imgs[a], z2_min, z2_max, n_z2,
Lambdas[a]))

OnFocus_imgs = []
for a in range(len(Cropped_imgs)):

OnFocus_imgs.append(np.uint8(np.absolute(U(Cropped_imgs[a],z2[a],Lambda
s[a]))))

# -------- Alignment - On Focus images ------------ #
```

```
Aligned_imgs = []
Aligned_imgs.append(Original_imgs[0])
for i in range(len(OnFocus_imgs)):
    if i != 0:

Aligned_imgs.append(alignment(OnFocus_imgs[0],OnFocus_imgs[i],Original_
imgs[i]))


# ****** Ploting On Focus (Cropped) images **********
fig = plt.figure(figsize=(10, 7))

fig.add_subplot(1, 3, 1)            # (rows, columns, position)
plt.imshow(OnFocus_imgs[0],cmap='gray')
plt.axis('off')
plt.title("Red")

fig.add_subplot(1, 3, 2)            # (rows, columns, position)
plt.imshow(OnFocus_imgs[1],cmap='gray')
plt.axis('off')
plt.title("Green")

fig.add_subplot(1, 3, 3)            # (rows, columns, position)
plt.imshow(OnFocus_imgs[2],cmap='gray')
plt.axis('off')
plt.title("Blue")

plt.show()


# -------- Multiespectral Phase Recorvering ------------ #
# --------- Cropping and Aligned holograms ------------- #
I = []
for i in range(len(Aligned_imgs)):
    I.append(Aligned_imgs[i])

# -------- Propagation - Cropped images ------------- #

# image to be used as a reference for functions of propagation
img = Original_imgs[0]
[Nx,Ny] = img.shape

# Frequence domain matrix
f = np.linspace(0,0+Nx, Nx,endpoint=False)
fxv = np.fft.ifftshift((f-np.floor(Nx/2+1))*(1/(Nx*dx)))
f = np.linspace(0,0+Ny, Ny,endpoint=False)
fyv = np.fft.ifftshift((f-np.floor(Ny/2+1))*(1/(Ny*dx)))
fx, fy = np.meshgrid(fxv, fyv, sparse=False, indexing='ij')

hU = []
U0 = []
```

```
U1 = [0,0,0]
phi_elem = [0,0,0]
phi_new = [0,0,0]
m_new = np.zeros((Nx,Ny))
phi = []
m = []

for a in range(len(I)):
    hU.append(np.sqrt(I[a]))
    U0.append(U(hU[a],z2[a], Lambdas[a]))
    phi.append(np.angle(U0[a]))
    m.append(np.abs(U0[a]))


print("Starting multispectral imaging processing")
print(".............................................")

for i in range(10):
    print("iteration: ", i+1)

    for a in range(len(I)):
        phi_elem[a] = phi[a]*Lambdas[a]
        m_new = (m_new + (1/3)*m[a])
    for a in range(len(I)):
        phi_new[a] =
(1/(3*Lambdas[a]))*(phi_elem[0]+phi_elem[1]+phi_elem[2])
    for a in range(len(I)):
        U0_new = m_new*np.exp(1j*phi_new[a])
        U1[a] = U(U0_new, z2[a], Lambdas[a])
        U1[a] = U(U1[a], -z2[a], Lambdas[a])
        phi[a] = np.angle(U1[a])
        m[a] = np.abs(U1[a])

img_R = np.absolute(U1[0])
img_G = np.absolute(U1[1])
img_B = np.absolute(U1[2])

Hologram_img = Cropped_imgs[2]
Amplitude_img = img_B
Phase_img = (np.angle(U1[2]))

# ****** Ploting Final Cropped images **********
fig = plt.figure(figsize=(10, 7))

fig.add_subplot(1, 3, 1)            # (rows, columns, position)
plt.imshow(Hologram_img,cmap='gray')
plt.axis('off')
plt.title("Hologram")

fig.add_subplot(1, 3, 2)            # (rows, columns, position)
plt.imshow(Amplitude_img,cmap='gray')
```

```
plt.axis('off')
plt.title("Amplitude")

fig.add_subplot(1, 3, 3)             # (rows, columns, position)
plt.imshow(Phase_img,cmap='gray')
plt.axis('off')
plt.title("Phase")

plt.show()


# Running time
end = time.time()   ###################################
print("\n""Running time: ", (end-start)//60, " min and
",round((end-start)%60,0)," s")

cv2.imwrite('Amplitude_final.png',rescale_intensity(Amplitude_img))
cv2.imwrite("Phase_final.png",rescale_intensity(Phase_img))
```

ANNEX D – PYTHON CODE TO CONTROL DEVICES FOR MULTISPECTRAL METHOD

```
"""
*** Python code to control the multispectral lens-free microscope ***
    Control of the camera (Arducam MT9J001) and RGB LEDs
"""

# ------------- Libraries  --------------------- #
import sys
import time
import os
import threading
from sys import exit

import cv2
import json
from ImageConvert import convert_image
import ArducamSDK

import serial

from datetime import date

global cfg,handle,running,Width,Heigth,saved_flag,color_mode,save_raw,
rtn_val, led_flag, z, folder, exposure, exposure_time, ser, z_new


# ---------- CONTROL PARAMETERS -------------- #
z = 0                 ## Auxiliary variable to control the image
acquisition
z_new = 0             ## Auxiliary variable to control saving images
led_flag = False      ## False if LED is off and can be used
running = True        ## True if the program is running
saved_flag = False    ## True to save the current image
flag_capture_error = False
cfg = {}
handle = {}
prp_exp = (100/31)    ## Proportion between exposure values in code and
time
intensity = 1

# ---------- ADJUSTABLE PARAMETERS --------------- #
save_raw = False            ## True to save file .raw of images
n_pictures = 3              ## Number of images
exposure = [320, 840, 110]       ## exposure values for PLASTIC DIFUSOR
exposure_time = [i *prp_exp*(1/1000) for i in exposure] ## exposure
time values in seconds

# Current address
addr = os.getcwd()

# ------------- Functions --------------------- #
# Set the configurations of the camera board
```

```
def configBoard(fileNodes):
    global handle
    for i in range(0,len(fileNodes)):
        fileNode = fileNodes[i]
        buffs = []
        command = fileNode[0]
        value = fileNode[1]
        index = fileNode[2]
        buffsize = fileNode[3]
        for j in range(0,len(fileNode[4])):
            buffs.append(int(fileNode[4][j],16))

ArducamSDK.Py_ArduCam_setboardConfig(handle,int(command,16),int(value,1
6),int(index,16),int(buffsize,16),buffs)

pass

# Write on configuration camera registers
def writeSensorRegs(fileNodes):
    global handle
    for i in range(0,len(fileNodes)):
        fileNode = fileNodes[i]
        if fileNode[0] == "DELAY":
            time.sleep(float(fileNode[1])/1000)
            continue
        regAddr = int(fileNode[0],16)
        val = int(fileNode[1],16)

        if regAddr != 12306: ## Add the desired exposure value
           ArducamSDK.Py_ArduCam_writeSensorReg(handle,regAddr,val)
        else:

ArducamSDK.Py_ArduCam_writeSensorReg(handle,regAddr,exposure[2])

pass

# Read the JSON file and set the parameters
def camera_initFromFile(fialeName):
    global cfg,handle,Width,Height,color_mode,save_raw
    #load config file
    config = json.load(open(fialeName,"r"))

    camera_parameter = config["camera_parameter"]
    Width = int(camera_parameter["SIZE"][0])
    Height = int(camera_parameter["SIZE"][1])

    BitWidth = camera_parameter["BIT_WIDTH"]
    ByteLength = 1
    if BitWidth > 8 and BitWidth <= 16:
        ByteLength = 2
        save_raw = True
```

```
    FmtMode = int(camera_parameter["FORMAT"][0])
    color_mode = (int)(camera_parameter["FORMAT"][1])
    #print("color mode",color_mode)

    I2CMode = camera_parameter["I2C_MODE"]
    I2cAddr = int(camera_parameter["I2C_ADDR"],16)
    TransLvl = int(camera_parameter["TRANS_LVL"])
    cfg = {"u32CameraType":0x4D091031,

            "u32Width":Width,"u32Height":Height,
            "usbType":0,
            "u8PixelBytes":ByteLength,
            "u16Vid":0,
            "u32Size":0,
            "u8PixelBits":BitWidth,
            "u32I2cAddr":I2cAddr,
            "emI2cMode":I2CMode,
            "emImageFmtMode":FmtMode,
            "u32TransLvl":TransLvl }

    # ArducamSDK.
    #ret,handle,rtn_cfg = ArducamSDK.Py_ArduCam_open(cfg,0)
    ret,handle,rtn_cfg = ArducamSDK.Py_ArduCam_autoopen(cfg)
    if ret == 0:

        #ArducamSDK.Py_ArduCam_writeReg_8_8(handle,0x46,3,0x00)
        usb_version = rtn_cfg['usbType']
        #print("USB VERSION:",usb_version)
        #config board param
        configBoard(config["board_parameter"])

        if usb_version == ArducamSDK.USB_1 or usb_version ==
ArducamSDK.USB_2:
            configBoard(config["board_parameter_dev2"])
        if usb_version == ArducamSDK.USB_3:
            configBoard(config["board_parameter_dev3_inf3"])
        if usb_version == ArducamSDK.USB_3_2:
            configBoard(config["board_parameter_dev3_inf2"])

        writeSensorRegs(config["register_parameter"])

        if usb_version == ArducamSDK.USB_3:
            writeSensorRegs(config["register_parameter_dev3_inf3"])
        if usb_version == ArducamSDK.USB_3_2:
            writeSensorRegs(config["register_parameter_dev3_inf2"])

        rtn_val,datas =
ArducamSDK.Py_ArduCam_readUserData(handle,0x400-16, 16)
        '''print("Serial:
%d%d%d%d-%d%d%d%d-%d%d%d%d"%(datas[0],datas[1],datas[2],datas[3],
```

```
datas[4],datas[5],datas[6],datas[7],

datas[8],datas[9],datas[10],datas[11]))
        '''
        return True
    else:
        print("open fail,rtn_val = ",hex(ret))
        return False

pass

def error_capture():

    global running_capture, running_acquisition, running_read,
error_capture_flag
    running_capture = False
    running_acquisition = False
    running_read = False
    #led_flag = False

    if lt.is_alive():
        lt.join()

    #Turn off the camera
    #close_camera()
    close_components()
    time.sleep(5)
    #Turn on the camera
    #init_camera()
    init_components()
    #led_flag = True

    if not lt.is_alive():
        lt.start()
    if lt.is_alive():
        lt.join()

    running_capture=True


# Perform image capture and add FIFO
def captureImage_thread():
    global handle,running, running_capture, flag_capture_error, running

    running_capture = True

    rtn_val = ArducamSDK.Py_ArduCam_beginCaptureImage(handle)
    if rtn_val != 0:
        print("Error beginning capture, rtn_val = ",rtn_val)
        running = False
```

```
        return
    else:
        print("Capture began, rtn_val = ",rtn_val)


    while running_capture:
        #print "capture"
        rtn_val = ArducamSDK.Py_ArduCam_captureImage(handle)
        if rtn_val > 255:
            flag_capture_error = True
            print("Error capture image, rtn_val = ",rtn_val)
            if rtn_val == ArducamSDK.USB_CAMERA_USB_TASK_ERROR:
                break
        else:
            flag_capture_error = False


    running_capture = False
    ArducamSDK.Py_ArduCam_endCaptureImage(handle)



# Get the image from FIFO, perform the conversion, and show the image
def readImage_thread():
    global
handle,running,Width,Height,saved_flag,cfg,color_mode,save_raw, folder,
addr, intensity, z, running_read, z_new
    global
COLOR_BayerGB2BGR,COLOR_BayerRG2BGR,COLOR_BayerGR2BGR,COLOR_BayerBG2BGR
    data = {}

    cv2.namedWindow("Lens Free",1)

    running_read = True

    while running_read:

        if ArducamSDK.Py_ArduCam_availableImage(handle) > 0:
            rtn_val,data,rtn_cfg =
ArducamSDK.Py_ArduCam_readImage(handle)
            datasize = rtn_cfg['u32Size']

            if rtn_val != 0:
                print(">>>>> Read data fail!")
                continue

            if datasize == 0:
                continue

            image = convert_image(data,rtn_cfg,color_mode)
```

```
            today = date.today()
            # Month abbreviation, day and year
            current_date = today.strftime("%b-%d-%Y")
            timestr = time.strftime("%Y%m%d-%H%M%S")
            ## save image in the folder
            z_str = '%d'%z_new
            z_str = z_str.zfill(4)
            if saved_flag:
                if not
os.path.exists(addr+"\\images_multispectral\\"+current_date+"_"+folder+
"_int%.2f"%intensity):

os.makedirs(addr+"\\images_multispectral\\"+current_date+"_"+folder+"_i
nt%.2f"%intensity)
                #path
=addr+"\\images_multispectral\\"+current_date+"_"+folder+"_int%.2f"%int
ensity+"\\image%d"%z_new+"-"+timestr+".jpg"
                path =
addr+"\\images_multispectral\\"+current_date+"_"+folder+"_int%.2f"%inte
nsity+"\\image"+z_str+"-"+timestr+".jpg"
                if(cv2.imwrite(path, image)):
                    print("Image saved successfully!!\n")
                    z_new+=1

                else:
                    print(">>>>> ERROR to save image.\n")

                saved_flag = False

            image = cv2.resize(image,(640,480),interpolation =
cv2.INTER_LINEAR)

            cv2.imshow("Lens Free",image)
            cv2.waitKey(5)
            ArducamSDK.Py_ArduCam_del(handle)

        else:
            time.sleep(0.001);

# Check the USB port that connects the LEDs to the computer
def check_usb_port():

    active_ports = []
    for number in ['COM%s' % (i + 1) for i in range(32)]:
        try:
            check_obj = serial.Serial(number)
            active_ports.append((number))
            check_obj.close()

        except serial.SerialException:
```

```
            pass
    return active_ports

# Perform the LEDs changing and the image acquisition control
def LED_thread():

    global saved_flag, running_acquisition, z, led_flag, ser,
n_pictures, flag_capture_error

    running_acquisition = True
    while running_acquisition:
        if led_flag == True:

            if flag_capture_error:
                break

            if(z<(n_pictures)):

                if not saved_flag:
                    if(z<n_pictures):
                        # Turn on only one LED color
                        if z==0:

ArducamSDK.Py_ArduCam_writeSensorReg(handle,12306,int(exposure[0]))

#ArducamSDK.Py_ArduCam_writeSensorReg(handle,12343,int(3))
                            ser.write(b'R')
                            time.sleep(2)
                            saved_flag = True
                            print('Image ', z+1,' - LED: Red')
                            time.sleep(5)
                        if z==1:

ArducamSDK.Py_ArduCam_writeSensorReg(handle,12306,int(exposure[1]))
                            ser.write(b'G')
                            time.sleep(2)
                            saved_flag = True
                            print('Image ', z+1,' - LED: Green')
                            time.sleep(5)
                        if z==2:

ArducamSDK.Py_ArduCam_writeSensorReg(handle,12306,int(exposure[2]))
                            ser.write(b'B')
                            time.sleep(2)
                            saved_flag = True
                            print('Image ', z+1,' - LED: Blue')
                            time.sleep(5)
                    ser.write(b'X')
                    saved_flag = False
                    z+=1
            else:
```

```
                led_flag = False # Turn off the LED
                z=0
                running_acquisition = False


def init_components():

    global config_file_name, ci, rt, lt, ser, flag_erro,
flag_capture_error

    # Start the LEDs
    print("Starting components: \n")
    print('--- LED RGB --- \n')
    try:
        # Start the LED
        port=check_usb_port()
        ser = serial.Serial(port[0], 9600, timeout=1)
        print('> Status: OK!')

    except:
        print('>>>>> Status: Error!')
        print(' LED not found')
        #print('Please, check if the LED is plugged in or if it is
connected to the computer.\n')
        flag_erro = True

    # Start the camera
    print('\n--- Camera ---\n')
    ser.write(b'L')     # Connect the camera to the computer
    time.sleep(2)
    if camera_initFromFile(config_file_name):
        ArducamSDK.Py_ArduCam_setMode(handle,
ArducamSDK.CONTINUOUS_MODE)
        print('> Status: Ok!\n')
    else:
        print('>>>>> Status: Error!')
        print(" Please, check if the camera driver has been installed
correctly or if the camera is connected.\n")
        flag_erro = True

    if (flag_erro):
        exit()

    # Create the threads
    ci = threading.Thread(target = captureImage_thread)
    rt = threading.Thread(target = readImage_thread)
    lt = threading.Thread(target = LED_thread)

    # Start the capture and read threads
    ci.start()
    rt.start()
```

```
def close_components():

    global running_capture, running_acquisition, running_read

    running_capture = False
    running_acquisition = False
    running_read = False

    global ci, rt, lt, ser
    # Close the threads
    ci.join()
    rt.join()

    cv2.destroyAllWindows()
    cv2.waitKey(1)

    print("Closing components:\n")

    rtn_val = ArducamSDK.Py_ArduCam_close(handle)
    if rtn_val == 0:
        print('> Camera closed successfully!\n')
        ser.write(b'D')     # Disconnect the camera to the computer
        time.sleep(2)
    else:
        print('>>>>> ERROR to close camera.\n')

    try:
        ser.close()
        print('> LEDs closed successfully!\n')
    except:
        print(">>>>> ERROR to close LED!")

def menu():

    print("\n======= LENS FREE MICROSCOPY =======\n")
    print("Command options:\n")
    print("  I - Change intensity values")
    print("  P - Preview (with blue LED)")
    print("  S - Start multispectral acquisition")
    print("  Z - Start a sequence of multiwavelength acquisition with
defined interval")
    print("  C - Close program\n")

def parameters():

    global exposure_time, intensity

    print('\n. Number of images: %s' %n_pictures)
    print('. LED: RGB')
```

```
    print('. Intensity proportion: %.2f' %intensity)
    print('. Exposure time for each LED is: %.2f s (Red), %.2f s
(Green) and %.2f s (Blue)' %(exposure_time[0], exposure_time[1],
exposure_time[2]))
    print('-------------------------------------------------\n')

def countdown(t):

    while t:
        mins, secs = divmod(t, 60)
        timer = '> {:02d}:{:02d}'.format(mins, secs)
        print(timer, end="\r")
        time.sleep(1)
        t -= 1
    print('End of the acquisition process.\n')
    print('-------------------------------------------------\n')


# ------------ Main function -------------------- #
def main():

    global running, led_flag, folder, exposure, exposure_time, ser,
intensity, saved_flag, config_file_name, flag_erro, n_pictures,
flag_capture_error, time_preview

    flag_erro = False

    #print("\n======= LENS FREE MICROSCOPY =======\n")

    ## Adress of JSON
    config_file_name = addr+"\\Json\\MT9J001_10MP.json"

    if not os.path.exists(config_file_name):
        print("File JSON not found.")
        exit()

    #init_components()

    while running:

        if not led_flag:
            time.sleep(1)
            menu()

            input_kb = str(sys.stdin.readline()).strip("\n")

            if input_kb == 'i' or input_kb == 'I': ## Time exposure
adjust for the camera
                print('Given the defaut exposure time for each LED is
%.2f s (Red), %.2f s (Green) and %.2f s (Blue).'%(exposure_time[0],
exposure_time[1], exposure_time[2]))
```

```
                intensity = float(input("Insert the intensity
proportion value: "))
                exposure_time = [i * intensity for i in exposure_time]
                exposure = [round(i *1000/prp_exp) for i in
exposure_time]

ArducamSDK.Py_ArduCam_writeSensorReg(handle,12306,int(exposure[2]))

            elif input_kb == 'p' or input_kb == 'P': ## Preview image -
turn on blue LED
                init_components()
                time.sleep(2)
                if not flag_capture_error:
                    time_preview = int(input("Insert time (s) for
preview:"))
                    ser.write(b'B')
                    time.sleep(time_preview)
                    ser.write(b'X')

                    #if lt.is_alive():
                    #    lt.join()
                close_components()

            elif input_kb == 'c' or input_kb == 'C': ## Close the
program
                running = False

            elif input_kb == 'z' or input_kb == 'Z': ## Multiple
acquisitions
                folder = str(input("Insert folder name:"))
                num_acquisitions = int(input("Insert the number of
acquisitions:"))
                time_interval = float(input("Insert the time interval
(min) between acquisitions:"))
                n=0
                while n< num_acquisitions:

                    if n==0:
                        init_components()
                    if not n==0:
                        init_components()


print('-------------------------------------------------')
                    print("[Iteration %d] Acquisition started with the
following parameters:" %(n+1))
                    parameters()
                    time.sleep(2)

                    start_acquisition = time.time()
                    led_flag = True
```

```
                    if flag_capture_error:
                        print(">>>>> FLAG_CAPTURE_ERROR")
                        n = n - 1

                    if not lt.is_alive():
                        lt.start()
                    if lt.is_alive():
                        lt.join()

                    #close_components()

                    if not n==(num_acquisitions - 1):
                        close_components()

                        if flag_capture_error:
                            pass

                        else:
                            end_acquisition = time.time()
                            time_aquisition = end_acquisition -
start_acquisition
                            time_interval_seg = time_interval*60 -
time_aquisition
                            print('Please, wait for more %.1f s'
%time_interval_seg)
                            countdown(int(time_interval_seg))
                    n=n+1
                close_components()

            elif (input_kb == 's' or input_kb == "S"): ## Single
acquisition
                init_components()
                time.sleep(2)

                if not flag_capture_error:
                    folder = str(input("Insert folder name:"))

print('-----------------------------------------------')
                    print("Acquisition started with the following
parameters:")
                    parameters()
                    time.sleep(2)
                    led_flag = True

                    if not lt.is_alive():
                        lt.start()
                    if lt.is_alive():
                        lt.join()
                close_components()
```

```
    '''if lt.is_alive():
        lt.join()'''

    #close_components()

if __name__ == '__main__':
      sys.exit(main())
```