A Study of Digital In-Line Holographic Microscopy for Malaria Detection

Elin Lundin
Carl Christian Kirchmann
Jakob Andrén
Abstract

A Study of Digital In-Line Holographic Microscopy for Malaria Detection

Andrén, Kirchmann, Lundin

The main purpose of the project was to create an initial lab set-up for a digital in-line holographic microscope and a reconstruction algorithm. Different parameters including: light source, pin-hole size and distances pinhole-object and object-camera had to be optimized. The lab set-up is to be developed further by a master student at the University of Nairobi and then be used for malaria detection in blood samples. To acquire good enough resolution for malaria detection it has been found necessary to purchase a gray scale camera with smaller pixel size. Two different approaches, in this report called the on-sensor approach and the object-magnification approach, were investigated. A reconstruction algorithm and a phase recovery algorithm was implemented as well as a super resolution algorithm to improve resolution of the holograms. The on-sensor approach proved easier and cheaper to use with approximately the same results as the object-magnification method. Necessary further research and development of experimental set-up was thoroughly discussed.
A Study of Digital In-Line Holographic Microscopy for Malaria Detection

Carl Christian Kirchmann, Elin Lundin, Jakob Andrén

August 12, 2014

Bachelor Diploma Project
Supervisor: Cris Luengo
Department of Physics, University of Nairobi
Department of Engineering Science, Uppsala University

Abstract

The main purpose of the project was to create an initial lab set-up for a digital in-line holographic microscope and a reconstruction algorithm. Different parameters including: light source, pin-hole size and distances pinhole-object and object-camera had to be optimized. The lab set-up is to be developed further by a master student at the University of Nairobi and then be used for malaria detection in blood samples. To acquire good enough resolution for malaria detection it has been found necessary to purchase a gray scale camera with smaller pixel size. Two different approaches, in this report called the on-sensor approach and the object-magnification approach, were investigated. A reconstruction algorithm and a phase recovery algorithm was implemented as well as a super resolution algorithm to improve resolution of the holograms. The on-sensor approach proved easier and cheaper to use with approximately the same results as the object-magnification method. Necessary further research and development of experimental set-up was thoroughly discussed.
We would like to give our thanks to
Ernst van Groningen, Kenneth Kaduki, David Karibe.
Conventions to be used in the report

\( d_p \): pinhole diameter
\( \theta \): angular resolution
\( \lambda \): wavelength
\( z_1 \): pinhole-object distance
\( z_2 \): object-camera distance
\( z_1 + z_2 \): pinhole-camera distance

Analog resolution: how well the features of the hologram are resolved, same definition as for diffraction.

Digital resolution: how well the analog resolution is discretized, always smaller or equal to the analog resolution.

DIHM: Digital In-Line Holographic Microscopy

FOV: Field of view
1 Introduction

1.1 Background

Malaria is a common blood disease that humans and other animals suffer in tropic and subtropic regions. It is a parasite that is transmitted by the female Anopheles mosquito. The parasite attacks the red blood cells which entails among others respiratory distress, cyclic fever and headache. It was estimated by the WHO that in 2009 there were 225 million cases of malaria, of which 781000 were fatal[2].

1.2 Identification of malaria

Microscopy is the most common way to identify malaria in red blood cells. First the blood sample is dyed, which takes approximately 30 minutes. Thereafter the dyed sample is examined manually with a microscope by personnel with sufficient knowledge and skills. Due to the limited field of view of the microscope the examination is rather time consuming. At least 100 microscopic images at 100× objective magnification are required to be manually scanned and the infected blood cells in each to be counted. According too [1] a normal field of view for a 100× objective magnification is about 0.18mm. Sampling 100 views with the side length of 0.18mm in a square results in an total minimum field of view of 1.8 × 1.8mm.

Another method of detecting malaria is by adding a certain antigen to the blood sample. This method only determines if the sample has malaria infected blood cells in it or not, without specifying the quantity. Due to the fact that most people in the tropics have malaria infected blood cells to some extent this is not a sufficient test. Often it is vital to know to what extent a person is infected. The second mentioned method leads to over-diagnosis since people who have only a few infected cells are diagnosed with malaria.

Due to lack of experience, skill, money or equipment many hospitals and clinics are not able to perform these methods of detecting malaria.

1.3 Purpose

It is clear that there is a vast need for a fast and cost efficient method of detecting malaria. An approach to make this more efficient is by substituting the personnel looking at the microscope samples with a device that analyses the samples and counts the percentage of infected blood cells. A way to make it cheaper is to replace the optical microscope with a digital in-line holographic microscope. This device is made up of: a LED or a laser, a pinhole and a camera. All this equipment is easily obtained for a much lower price than a conventional optical microscope.

A digital in-line holographic microscope for field diagnosis of malaria will be developed at the university of Nairobi as subject for a master thesis by David Karibe. With a major in embedded systems, he will integrate the reconstruction of the holograms in a field
programmable gate array for more efficient computations and so that the device will not need a computer to analyse the samples. The purpose of our bachelor degree project is to build a prototype of a digital in-line holographic microscope and carry out experiments to investigate how different parameters in the set-up (such as light source, pinhole size and distances between object and screen and between pinhole and screen) will affect the resolution of the microscope. This will act as a foundation for David Karibes further research in the area.

1.4 Method

The first part of the project was to gain understanding of optical systems and holography. The fundamental concept of point spread function and its part in the mathematical description of light propagation were understood and are presented in Section 2. To further deepen our understanding of holography and light propagation numerical simulations were carried out on constructing simple holograms. Different methods to numerically propagate light were investigated in simulation (section 7.1) and in theory (section 6.1). Next, an experimental set-up should be built. Stability is a crucial factor in hologram recording and part of the project plan was to 3D-print a stable benchtop set-up, with a holder for camera, sample and light source respectively that would still allow distance parameters to be adjustable. Theory on 3D-printing is presented in 5.2 and the experimental set-up is described in 2. For 3D-modeling SketchUp was used. A microscope for malaria detection needs to be able to resolve details in the micrometer scale. The size of a blood cell is about 8 µm and to be able to resolve even smaller parts of the blood cell, such as the malaria parasite, the required resolution is around 1 µm or smaller. In order to maximize resolution researchers on digital in-line holographic microscopy have branched into two approaches of setting up the experiments. One approach take advantage of the built-in magnification that is attained when the sample is placed close to the pinhole and the distance $z_1$ is small compared to the object-camera distance $z_2$, whereas the other approach will have $z_2 \ll z_1$ to allow usage of a larger pinhole. The two different approaches were investigated in the project. Theory is presented in section 3 and results from experiments in section 7. To improve resolution beyond the limit of the camera pixel size, two different image super resolution algorithms were implemented in the project, both described in section 6.2. Also, to be able to use the full potential of the holographic technique, allowing the reconstruction of object phase as well as intensity, a phase reconstruction algorithm has been implemented. Methods to further improve resolution beyond the results in this project are suggested in section 9. Time was not enough to carry out these methods in practice.
Figure 1: Result of light with wavelength $\lambda = 635\text{nm}$ going through a circular aperture with diameter $50\ \mu\text{m}$. In order to see the fringes the hologram had to be overexposed.

2 Diffraction theory and the propagation of light

The word diffraction comes from Latin and means *to break into pieces*. This is what happens when a wave engages an aperture, slit or object in roughly the same size as its wavelength. The phenomenon is explained by the principle of superposition of waves and the Huygens-Fresnel principle. The principle of superposition of waves states that the net amplitude vector of many wave fronts are a sum of the independent waves amplitude vectors. When two waves interact in such a way that the net amplitude is increased it is called constructive interference and otherwise destructive interference.

Huygens-Fresnel principle is a useful method of interpreting a wave front. The principle declares that every point on a wave front can be seen as a new point source. A point source emanates waves in all directions spherically. When applying this to a wave front engaging a slit of similar size to its wavelength, in addition to the principle of superposition, one can calculate the resulting wave front at a certain distance from the slit\[7, 45-46].

In figure 1 the result of light which has gone through an aperture is seen and it is clearly seen that due to constructive and destructive interference one can see the diffraction pattern.
Mathematically, the calculation of a wave front in the image plane by summing the contribution from all points in a source plane, is expressed by the Fresnel-Kirchoff formula:

\[ A(x, y, z) = \frac{-ik}{2\pi} \iint_{\Omega} \frac{\exp(ikr)}{r} A_0(x_0, y_0) dx_0 dy_0 \]  

(1)

\[ r = \sqrt{(x - x_0)^2 + (y - y_0)^2 + z_2^2} \]  

(2)

where \( A_0(x_0, y_0) \) is the field in the source plane, \( \Omega_0 \), and \( A(x, y, z) \) the field at the image plane, \( \Omega \) \cite{24, 20} and \( r \) is defined in \cite{2} The integral can be expressed as a convolution of \( A_0 \) with the point spread function (psf) \( h \).

\[ A(x, y, z) = A_0 * h \]  

(3)

The psf is commonly used in imaging systems to describe how the light from a point source spreads out with propagation distance \( r \). The point spread function \( h \) is defined as:

\[ h = \frac{-ik \exp(ikr)}{2\pi r} \]  

(4)

That the Fresnel-Kirchoff formula is a convolution is of great importance in the holographic reconstruction process, where light has to be back-propagated from the holographic image to the object plane. By the convolution theorem, the Fourier transform of a convolution is the point wise product of the convolved functions’ Fourier transforms.

\[ \mathfrak{F}(f * g) = \mathfrak{F}(f) \mathfrak{F}(g) \]  

(5)

The propagation of light between two planes can then be efficiently calculated by use of the fast Fourier transform.
3 Digital in-line holography

The main difference from holography and other imaging techniques is holography’s ability to capture both information about the intensity distribution as well as the phase. In for instance photography only intensity is obtained. In conventional holography the phase information is captured by using coherent light to illuminate the object that is to be imaged. This coherent illumination is called the reference beam and is scattered by the sample and the information is collected by a sensor. Knowing that the reference beam is coherent one can extract phase information from the scattered beam collected by the sensor since the intensity also contains phase information. [37] 1-4].

A hologram is given by the interaction between the light that is scattered by the object, the object wave $A_{obj}$, and the coherent light beam emanating from the pinhole $A_{ref}$. The total wave amplitude is then a sum of the object wave and the reference wave $A = A_{obj} + A_{ref}$ and the intensity $I_{screen}$ that can be measured on the screen is given by

$$I_{screen} = A_{ref}^* A_{ref} + [A_{obj}^* A_{ref} + A_{ref}^* A_{obj}] + A_{obj}^* A_{obj}.$$  

(6)

The two terms in the brackets are the holographic terms, of the real and the virtual image respectively. The last term is the cross-interference term, which comes from the object’s interference with itself as in normal diffraction theory, and the first term is the background intensity from the reference beam [29]. To reconstruct the object image one would like the holographic terms to be dominant in the measured screen intensity. The object image can then be obtained as in analog holography by shining the same reference beam through a print of the hologram, or it can be done numerically. The digital, numerical, reconstruction process is described in Section 6.

When will the holographic terms dominate the intensity? For an object like the double slit, most of the reference beam will be blocked out and the object wave can only interfere with itself. The cross-interference term will dominate the image. On the other hand, if the object is very small or highly transparent, the object amplitude will be small compared to $A_{ref}$ and the holographic terms will dominate over the cross-interference term that is quadratic in $A_{obj}$ [29].

Furthermore, there are two holographic terms, but only one of them corresponds to the real image of the object. The other term is the cross-interference term, which is often referred to as the "twin image." This is a virtual image of the object, placed at the same distance $z_2$ from the screen as the real object, but at the opposite side of the screen. The twin image might cause problems in the reconstruction process, especially if the object is placed close to the recording sensor. The twin image is discussed in Section 6.3.

To extract the holographic terms one sometimes wants to remove the reference wave intensity. This can be done by simply subtract an image of the reference beam from the hologram. However if a reference image is not captured, it can be removed by processing the hologram with a high-pass filter. In that way all low frequency terms, and hence the reference beam, are suppressed. Parts of the image with small changes in color values correspond to low frequencies as opposed to greater changes, for example edges and noise, which correspond to high frequencies [35]. The image is filtered by creating a Gaussian
(a) High-pass filtered reconstructed image
(b) Reconstructed image from original hologram

Figure 2: Reconstructions with the use of a high-pass filter and without, a) and b) respectively. The filtered image, a), contains less of the reference beam that is prominent in the middle of the unfiltered image b).

matrix, a low pass filter, which then is convolved with the hologram. To only observe the higher frequencies the result of the convolution is subtracted from the original image, hence the low frequencies are removed from the hologram. The original reconstruction and the high-pass filtered reconstruction can be seen in fig 2.

3.1 Two approaches to digital in-line holographic microscopy

When in-line holography is used as a microscopy method, there are two rather different approaches that has been proven successful. The oldest of the two involves the use of a fully coherent light source, such as a laser. The beam from the light source is then forced through a small aperture ($d_p \sim 1 - 2 \mu m$), before it illuminates the object placed at a small distance ($z_1 \sim 1 - 2 \text{ mm}$) from the pinhole. The camera is positioned a few centimeters away, thus creating a magnified hologram of the object.

The other approach has been developed in recent years, with great contribution from the Ozcan group [21][10][31][9][30][42]. Instead of using a laser, advantage is taken on partially coherent light sources such as LEDs. The pinhole size is big ($d_p \sim 50 - 100 \mu m$) and the object is now placed very close to the camera rather than to the pinhole, while the pinhole-camera distance is rather long, $z_1 + z_2 \sim 40 - 100 \text{ mm}$. The large distance $z_1$ allows the light beam to gain sufficient spatial coherence to illuminate individual micrometer sized objects in the object plane with spatially coherent light [21].

In the first approach a small pinhole is essential to achieve high resolution. Small pinholes with diameters of $d_p = 1 - 2 \mu m$ clog easily (from experience we know that a 10 $\mu m$ pinhole can be ruined entirely by a fingerprint) and requires bright and well-focused lasers. The advantage of bigger pinholes are that the set-up can be made more compact when no
focusing is required, that less intense and hence cheaper laser diodes or LEDs can be used and that the set-up is less sensitive to misalignment. When the object is placed more or less on top of the recording sensor the field of view (FOV) is also significantly bigger than for the object-magnification approach and, in fact, also much bigger than for a normal lens microscope. The on-sensor approach, on the other hand, has the drawback that it requires more computational power and numerical processing of the recorded holograms.

3.1.1 Object-magnification DIHM

The resolution of the coherent DIHM approach is governed by the numerical aperture, $NA$, of the set-up. When a beam of light is spatially filtered through a small pinhole the zero order diffraction pattern will propagate like a cone along the z-axis of the set-up. In in-line holography the sample volume should lie entirely within this light cone and to attain maximum resolution one would want this cone to be as big as possible. A measurable quantity of the zero order emission cone is its half angle $\theta_{cone}$ and we define the quantity $NA_{pinh}$ according to

$$NA_{pinh} = n \sin \theta_{cone} = 1.22 \frac{\lambda}{d_p}$$

where $n$ is the refractive index of the medium after the pinhole, $d_p$ is the pinhole diameter and $\lambda$ is the wavelength of the light beam[16]. It is clear that a smaller pinhole gives a wider emission cone and for this approach it is always desirable to use a pinhole with minimum possible diameter. However, the maximum possible recording angle $\theta_{cam}$ is important for how small details that will be possible to restore from a hologram. The related measure $NA_{cam}$ is defined as

$$NA_{cam} = n \sin \theta_{screen} = \frac{(W/2)^2}{\sqrt{(W/2)^2 + L^2}}$$

where $W$ is the camera screen width and $L = z_2$ is the distance from the object to the camera[29]. To maximize $NA_{cam}$, and hence the resolution, one would want to put the screen as close as possible to the object and to use a camera screen as large as possible. Normally, the distance $z_2$ between the object and the camera is still not smaller than 1.5 $-$ 3 cm to avoid aliasing in the recorded hologram. A minimum distance to avoid aliasing $z_{2,min}$ can be calculated as

$$z_{2,min} = \frac{\delta p}{\lambda} (W + L_{ob})$$

where $L_{ob}$ is the lateral size of the object and $\delta p$ is the pixel size[12]. Given that the screen cannot be moved too close to the object, the resolution will mainly be determined by the size of the camera sensor for a fixed pinhole and wavelength. This means that a more expensive camera will give better resolution. To increase resolution, another option is to decrease the wavelength, possible at the expense of a more costly laser.
The lateral resolution of a hologram can be estimated

\[ \delta_{\text{lat}} > \frac{\lambda}{2NA} \]  \hspace{1cm} (10)

where \( \delta_{\text{lat}} \) is the distance between two points in the object plane that can be just resolved. \( NA \) is the minimum quantity of \( NA_{\text{pinh}} \) and \( NA_{\text{cam}} \). A corresponding estimation of the axial resolution is

\[ \delta_{\text{ax}} > \frac{\lambda}{2NA^2} \]  \hspace{1cm} (11)

Since \( NA < 1 \) it is harder to resolve axial details than lateral details.

### 3.1.2 On-sensor DIHM

The on-sensor DIHM approach takes advantage of a big pinhole and partially coherent light source. This is possible since, when the object is placed close to the camera sensor, the distance \( z_1 \) is much bigger than \( z_2 \), and the illuminating aperture is demagnified by a factor \( z_1/z_2 \) in the same time as the coherence diameter of the reference light beam grows with distance. Putting the sample directly on top of the camera sensor also means that there is no magnification of the object and hence the full sensor size will constitute the FOV of the object.
Furthermore, the Ozcan group argue that the use of an incoherent light source reduces the influence of the cross-interference term from equation (6). For holographic reconstruction this term is regarded as speckle noise and the use of a partially coherent light source will as such improve the hologram quality [21][31][42].

The cross-interference term arises when object waves from different parts of the sample plane interfere with each other. However, the intensity of the cross-interference term from two objects which are further apart from each other than the coherence diameter in the object plane, will decrease rapidly with distance. In the object plane the coherence diameter will be proportional to $\lambda_0 z_1/d_p$ (where $\lambda_0$ is the mean wavelength of the incoherent light). A larger pinhole will inhibit the contribution of the cross-interference term for objects from different parts of the sample, whereas a small pinhole will light up a much bigger portion of the sample with coherent light and the hologram with a fully coherent source will contain much more cross-interference, so called speckle noise. This is of special importance in dense samples such as blood smear where the cross-interference term might be large. And as far as it goes to capturing the holographic terms of the objects in the sample, it holds that as long as each cell is smaller than the coherence diameter of the illuminating light in the object plane, the individual cells will be illuminated by coherent light and contribute to the hologram [31].

In the on-sensor approach, the object is placed very close to the camera and the discussion of resolution takes another shape than for the case of the object-magnification approach. For the short distance $z_2$, the detection NA, $N.A_{cam}$, approaches 1 and the resolution is now limited by the pixel size more than anything else. To move beyond this limit it is possible to record several, slightly shifted, holograms and from these construct a super-resolved hologram capable to resolve smaller fringes. This method goes under the name pixel super resolution and will be discussed in section 6.2.
4 Experimental set-up

In figure 4 a schematic of the experimental set-up for the on-sensor approach is seen. The pin-hole is attached closely to the LED and the pinhole is located a distance $z_1$ from the sample. The sample is then positioned a distance $z_2$ from the sensor. For capturing multiple translated low resolution holograms the sensor and sample together can be displaced in the $x-$ and the $y-$direction. To do phase recovery two or more holograms have to be captured at different $z_2$ and therefore the sample can be moved, independently of the sensor and pin-hole, in the $z-$direction.

Figure 5 shows the implementation of the schematic, figure 4. The micrometer screws allows displacement in the $x-$ and the $y-$direction and the DVD parts move the sample in the $z-$direction. The pin-hole is attached directly to the LED, though not visible in figure 5. The protective plastic was removed from the camera to be able to place the sample as close as possible to the camera. To transfer the images from the camera to the computer a USB cable was used. The software uEye Demo was used to configure exposure time, frame rate and switch between normal debayered and raw output.

The camera that was used has a pixel size of $2.8 \times 2.8 \mu m$ and 1600 $\times$ 1200 pixels, which gives a screen size of $3.4 \times 4.5 mm^2$. The dimensions of the camera are $36 \times 36 \times 20 mm$. It is a color camera with the most common color filter, the Bayer filter. In the Bayer Filter each pixel has a filter that enables only red, green or blue to pass, alternating as in figure 6. The side effect of this filter is the effective magnification of the pixel size by a factor of two, when using the blue (red) light source as in our case. This is due to the fact that all pixels that captured red (blue) or green do not have any useful information, only noise. When a monochrome camera sensor is used all the pixels will be used to capture the intensity of the incoming light.
Figure 5: The experimental set-up that was used.

Figure 6: Bayer filter pattern on image sensor. Source: Bayer filter [5]
The pinholes were all *Precision Pinholes* from Edmund Optics in sizes varying from 1 \(\mu m\) to 100 \(\mu m\) \[34\]. The source of illumination, the LED, emitted light with wavelength \(\lambda = 465\, nm\) and maximum power of 20\(mW\). The wavelength of the laser that was used is \(\lambda = 635\, nm\) and power of 90\(mW\) \[35\].

### 4.1 Using the set-up

The set-up was used as follows: first the sample was placed as close as possible to the sensor and all surrounding lights turned off. Using *uEye Demo* the exposure time, frame rate and output mode was configured. Thereafter approximately 20 holograms were captured, all of which were slightly shifted with the help of the micrometer screws. Next the sample was shifted 30 – 70 \(\mu m\) in the \(z\)–direction and the previous procedure was repeated. The shifting of the sample and sensor with the micrometer screws proved time consuming. As mentioned before this is only an experimental set-up and a proposal for a better, automatized set-up is discussed in section \[9\].
5 Constructing the lab set-up

Among the goals of this project was to build a stable lab set-up for experimentally recording holograms, using 3D-modeling and 3D-printing. SketchUp was used for 3D-modeling. Some rules and tips for using SketchUp are accounted for in Section 5.1. Theory of 3D-printing and how it was used in the project is presented in Section 5.2.

5.1 Modeling in SketchUp

There are some important rules and features in SketchUp one has to consider to be able to successfully model and 3D print an object. All faces must be closed on the object, meaning that it has to be a solid you are modelling. Also all walls must have a finite thickness that is thicker than the layer resolution of the machine. The model has to be manifold, this means that an edge cannot share more than two surfaces.

To ensure precision while modelling it is good to start with a model that is 100 or 1000 times larger than the intended model and then resize it. By doing this you avoid problems with missing faces and inaccuracy in your model. This problem has its origin in the fact that SketchUp initially was intended for architecture and therefore less than millimeter precision was not needed. To get smooth curves the number of arc segments of the curve can be increased.

Whenever the model contains parts which are not fully supported by something underneath them it is said to have overhang. To cope with this one has to add support materials underneath the part of the object that is without support or 3D print the object in such a way that there will not be any overhang, for instance rotating it before printing. Without support material the constructed 3D object may have unevenness. Another way of dealing with problems with overhang is to divide your object into different parts and print them separately and then glue them together, however the gluing may introduce new errors.

When all these rules and tips have been accounted for when modelling one can start to generate the G-code. G-code is the map for the extruder, it tells it where to extrude plastic, print speed, how much to extrude and at which temperature. In Google SketchUp 8 the plugin CADspan tools was used to convert the model into a stl file from which the software ReplicatorG could generate G-code.

The program used in this project was the freeware Google SketchUp 8 with the plugins Solid Inspector version 1.2.0 and CADspan.

5.2 3D printing

A 3D-printer is a machine that creates a 3D solid or hollow object from a digital model. The machine interprets a digital model as a series of cross sections. These cross sections are printed by adding hundreds or as many as thousands of thin layers of building material on top of each other to create a 3D shape. The printing process is executed by a machine
that can move in three linear directions and is powered by small stepper motors. The
stepper motors determine the positioning precision of the 3D printer [15, 1-5].

On these stepper motors an extruder is located, being responsible for the output of print-
ing material. Different materials can be used, such as polymers, metals and ceramics. When choosing material it is important to consider the application of the 3D printed
object since the materials have a variety of properties and price ranges. In the intended
experimental set-up there were no specific requirements and therefore plastic, being very
cheap, was used. In the case of plastic being used in fused deposition modelling the
filament is fed to a heating chamber within the extruder that heats the plastic to approx-
imately $200^\circ$. The heated plastic is then extruded through a opening with about $0.5mm$
radius [15, 2-3].

The surface that is used to print on is called the printbed and is made of differing materials
depending on 3D-printer. The printbed should have several features such as good heat
conductivity so that the material is cooled down quickly. Also they should have the
ability to keep the object from leaving the surface while 3D printing, this is often done
by putting a layer of tape on the printbed [15, 3-4].

The process of 3D printing can take from minutes to days depending on the resolution
of the machine, the print speed and the object that is printed. Our set-up took approx-
imately one and a half hour to print, with a $10\%$ fill ratio. The actual time for the
whole process of printing the set-up was nearly three hours due to problems with initi-
ating the printer and generating the G-code. In 3D printing resolution consist of both
layer resolution as well as positioning precision in space [28].

Due to rapid development over the last years, costs have been dramatically reduced
making it a viable option for production and prototyping in many areas, among others:
architecture, aerospace, military and engineering. The printer was available for use on
short notice and hence made it excellent for trying new ideas and prototypes. The printer
used in this project was a Replicator Dual with $100\mu m$ layer resolution, $11\mu m$ $x-$ and
$y-$axis positioning precision and $2.5\mu m$ $z-$axis positioning precision.
Aquire \( X_1 \) LR holograms at \( z_{2a} \)

Create SR hologram

Use autofocus to calculate \( z_{2b} \)

Phase recovery

Aquire \( X_2 \) LR holograms at \( z_{2b} \)

Create SR hologram

Use autofocus to calculate \( z_{2a} \)

Reconstruction

Figure 7: Flowchart of the process of DIHM with super resolution, intensity reconstruction and phase recovery.

6 The reconstruction process

The object described by a hologram can be reconstructed either in an analogue or a digital way. Analogue holograms that are captured on holography plates do not need any computing to be reconstructed, hence the name analogue holograms. To reconstruct analogue holograms the holography plate has to be illuminated by the reference beam from the same angle as it was recorded.

In digital holography, where a hologram is recorded with a camera sensor, reconstruction becomes computationally intense. An algorithm, based on the propagation of light and the concept of point spread function (section 2), has to be implemented. To increase object resolution beyond the limit of camera pixel size a super resolved hologram can be created from several low resolution holograms. The super resolution algorithm we have used in this project is described in Section 6.2.

In the recording process of a hologram, only intensity measurements are collected. In order to restore phase information an iterative process involving holograms captured at different recording distances \( z_2 \) has been implemented and is described in Section 6.3. To find the accurate recording distance with minimum user interaction we use an autofocus algorithm, described in Section 6.4. The entire process is displayed in figure 7.
6.1 Numerical reconstruction

The mathematical expression to back propagate the object wave from the hologram plane \( \Omega_0 \), a distance \( L \), to the object plane is given by the Fresnel-Kirchhoff formula \[8\]

\[
A(x, y) = \frac{i}{\lambda} \int \int_{\Omega_0} A_0(x_0, y_0) \exp \left( -\frac{ik\sqrt{(x - x_0)^2 + (y - y_0)^2 + L^2}}{(x - x_0)^2 + (y - y_0)^2 + L^2} \right) Ldx_0dy_0
\] (12)

The integral is a convolution of the wave front in the hologram plane, \( A_0 \), and the point spread function, here denoted \( h \):

\[
h(x, y) = \frac{i}{\lambda} \exp \left( -\frac{ik\sqrt{x^2 + y^2 + L^2}}{x^2 + y^2 + L^2} \right) L.
\] (13)

There are different methods to calculate the Fresnel-Kirchhoff integral. The Fresnel Transform Method described in [24] is based on the propagation of spherical wave fronts and requires only one fast fourier transform, but the pixel resolution depends on propagation distance. Also the Huygens convolution method is based on spherical wave fronts, but for this method the pixel resolution is fixed [24]. The Fresnel-Kirchhoff integral is calculated according to the convolution theorem and requires the calculation of three fourier transforms:

\[
A(x, y) = \mathcal{F}^{-1}(\mathcal{F}(A_0)\mathcal{F}(h))
\] (14)

This method is limited in its ability to accurately reconstruct objects for different recording distances \( L = z_2 \), both for small distances and large distances. The problem lies with the \( \frac{L}{x^2 + y^2 + L^2} \) factor, which is problematic to numerically Fourier transform for both small and large \( L \) [24]. This problem is remedied if the Fourier transform of the psf is interchanged with its analytical counterpart, as was suggested by Ayoub \textit{et al.}[8].

\[
\mathcal{F}(h) = \exp \left( -\frac{2\pi i}{\lambda} L \sqrt{1 - \lambda^2 k_x^2 - \lambda^2 k_y^2} \right)
\] (15)

where \( k_x \) and \( k_y \) are spatial frequencies. This method also goes under the name The Angular Spectrum method and is the method we have used in the project, due to its unlimited range in recording distance.

6.2 Super Resolution

Super resolution is a method for improving digital resolution in discretized images acquired by a sensor array of limited resolution. The quality of the captured image is affected by several parameters seen in figure 8 and include: optical distortion, motion blur,
Figure 8: From left: original image, optical distorted, motion blurred, aliased and with noise. Only the last two are present in DIHM.

aliasing and noise \[36\]. DIHM results are not degraded by optical distortions as DIHM does not include optics. Motion blur is not a problem as the setup is rigid and static when capturing the images. Aliasing is a major degrading factor for on-sensor DIHM as the physical pixel size of about \(2 - 3 \, \mu m\) can not resolve the sub micrometer fine details needed for malaria analysis. Depending on the sensor used and the intensity of light passing trough the pinhole, noise might as well become a problem if the signal to noise ratio (SNR) is to small. Super resolution works very well on improving aliased images and is able to lower the effective pixel size 6 fold, from \(2.3 \, \mu m\) to \(\sim 0.37 \, \mu m\) \[13\]. Super resolution also gives some noise suppression control if needed. This yields that super resolution may improve the digital resolution to be almost in pair with the analogue resolution.

6.2.1 Implementation

A super resolution image is formed by taking a sequence of low resolution (LR) images with slightly translated origins. The translations only matter on the sub-pixel scale and can be random, but preferably evenly distributed and with no duplicates. We performed these displacements by having our pinhole fixed pointing down. The camera with the sample very close, \(\sim 1 \, mm\), attached to a micrometer bench with which we could preform small stable movements. These images are then repositioned to the nearest integer pixel in the high resolution grid in which we want to reconstruct the final high resolution (HR) image in. The mean is calculated if to LR-images share the same coordinate.

The LR-images displacements has to be registered and be placed relative to each other with high accuracy. The error has to be less than 0.1 LR-pixels \[26\] for a good result. This may be done either by cross-correlation in Fourier-domain or a gradient based approach with the image. The Ozcan group uses the gradient based approach \[13, 18\] and it is determined in \[26\] that the gradient as an iterative method is the most accurate one to use. This method is implemented in the DIPimage toolbox\[27\] a competent Matlab library that we used in our implementation for registration of our images.

In figure 9 three LR images are super positioned with their displacement onto a HR grid. The displacements are rounded to the closest HR pixel and they are then placed on the HR grid. Because of the random displacements and a lower number of LR images than the number of new subpixels to fill, some of the HR pixels will remain empty after this procedure. In this case the up sampling factor is four which gives 16 new subpixels and
there is three LR images with no duplicates, which results in 13/16 HR pixels to be empty. These gaps can be filled by either using normalized convolution or minimizing a cost function. The normalized convolution calculates the convolution of the sparse HR grid with a gauss kernel and divides the result with the convolution of the sparse binary HR grid with the same kernel

\[ HR_{filled} = \frac{HR_{sparse} \ast g}{c \ast g}, \quad c = \begin{cases} 1, & \text{if } HR_{sparse} > 0 \\ 0, & \text{if } HR_{sparse} = 0 \end{cases} \]

(16)

This is easy to implement and fast to execute but the results are poor in comparison with the cost minimization method described below. \[7.6\]

The cost minimization method gives better results and operates by estimating a HR image that fits the sequence of LR images as well as possible. The estimation is done with the observation model seen in \[10\] and defines how the LR and HR image corresponds to each other. One pixel in the LR image is the sum of the HR pixels that it bounds, seen as a grayed area in the picture. A HR pixel corresponds to the sum of the overlapping LR pixels. For the mathematical definition please see \[13\] \[20\].

The HR image estimate is defined by the \( \hat{z} \) that minimizes the cost-function derived from the observation model.

\[
C(z_k) = \frac{1}{2} \sum_{m=1}^{pM} \left( y_m - \sum_{r=1}^{N} w_{m,r} z_r \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{N} \left( \sum_{j=1}^{N} \alpha_{i,j} z_j \right)^2
\]

(17)

where \( z_k \) is a pixel in HR. The sums corresponds to convolutions, \( w_{m,r} \) and \( \alpha_{i,j} \) is then seen as convolution kernels. \( w_{m,r} \) is a square of ones divided by the number of elements in \( w_{m,r} \), with the same width as the up sampling number. \( \alpha_{i,j} \) is the two dimensional discrete Laplace operator convolved by itself. The second part of the cost function is a smoothing
term necessary for finding a unique global minimum. The smoothing term penalizes high
frequency features, normally noise, in the HR image this forces the function to become
well behaved. The cost function above is the definition for a one dimensional vector but
the same is true for two dimensions. The cost function is minimized by finding the zero
point for the partial derivative

\[ g_k(z_k) = \frac{\partial C(z_k)}{\partial z_k} = \sum_{m=1}^{pM} w_{m,k} \left( \sum_{r=1}^{N} w_{m,r} z_r - y_m \right) + \lambda \sum_{i=1}^{N} \alpha_{i,k} \left( \sum_{j=1}^{N} \alpha_{i,j} z_j \right) \]  

and iterating

\[ \hat{z}_k^{n+1} = \hat{z}_k^n - \varepsilon^n g_k(\hat{z}_k^n) \]  

where

\[ \varepsilon^n = \frac{\sum_{m=1}^{pM} \gamma_m \left( \sum_{r=1}^{N} w_{m,r} \hat{z}_r^n - y_m \right) + \lambda \sum_{i=1}^{N} \bar{g}_i \left( \sum_{j=1}^{N} \alpha_{i,j} \hat{z}_j^n \right)}{\sum_{m=1}^{pM} \gamma^2_m + \lambda \sum_{i=1}^{N} \bar{g}^2_i} \]  

\[ \gamma_m = \sum_{r=1}^{N} w_{m,r} g_r(\hat{z}_n) \]  

\[ \bar{g}_i = \sum_{j=1}^{N} \alpha_{i,j} g_j(\hat{z}_n) \]  

Figure 10: Observation model between the LR and HR image.
Figure 11: Fictional iteration procedure for the cost minimazation.

\[ \text{Figure 11 represents a fictional procedure for the iteration where the cost function for a pixel } z_k \text{ intensity is the blue line and the iterate path is the red. The iteration continues until the difference between the previous and new iteration step is smaller than a given threshold value or a set maximum number of iterations are completed. } \epsilon \text{ and } g_k(z_k) \text{ both goes towards zero as the solution converges.} \]

See Appendix for example code, function sr_cost.m. For further reading see [13, 26, 20] for implementation of super resolution and [39] for how to solve the cost minimization.

### 6.2.2 Computational cost

Super resolution is a computational intense approach for acquiring higher digital resolution. This is especially true for on-sensorDIHM because of the extremely large field of view. The number of pixels \( n^2 \) for a theoretical \( n \) times improvement of resolution when field of view is kept constant. Optimally the same number of LR-images are acquired for a full sampled signal. A theoretical approach for determine the number of HR-pixels needed, is to study the Nyquist criterion. It states that you need more than two samples, in our case pixels, per feature period. A 1 \( \mu m \) resolution then needs pixels smaller than 0.5 \( \mu m \), which if you use our camera with dimensions 3.36 \times 4.48 \text{ mm yields a HR-image with dimensions 6720 \times 8960 pixels, about 60 megapixels. In reality this is probably not enough due to noise and blurring caused by the large sampling area of each pixel. This might become a problem for future implementations of stand alone malaria detection systems where computational power might become the most expensive part of the equipment. A way to work around this is to crop the field of view, either digitally or by choosing a smaller sensor.}

The image registration of the LR-images is as well computationally demanding. The algorithm used iterates small shifts until the correct position is found. This has proven
slow and cropping of the image has been done for acquiring faster results.

The number of iterations needed by the cost minimization rises with the up sampling factor, with the number of LR images constant. Each iteration is more demanding for higher up sampling and we would therefore recommend to only up sample as much as needed and only the area of interest to minimize computational cost.

6.3 Phase reconstruction

When a hologram is recorded, the phase cannot be measured directly. However, information about the phase is encoded in the intensity variations of the recorded hologram. In this project we use a phase recovery method developed by the Ozcan group [9] to extract phase information from intensity measurements.

6.3.1 Advantages of phase reconstruction

A hologram consists of both the real image of an object and the virtual, ”twin”, image positioned at the same distance on the opposite side of the camera relative to the real image (remember the two holographic terms of equation (6)). In the reconstructed real image, the twin image will be seen as a blurry background and when dense samples are considered in the on-sensor approach (e.g. a short distance between the object plane and the twin object plane) this can considerably reduce the quality of the hologram. Phase recovery can help reduce the impact of the twin image artefact [9] and improve the quality of the reconstructed image.

Also, and perhaps most importantly, the phase image of an object can contain information about the object that is not captured by the intensity image. The goal of this project is to provide information on holographic microscopy that can be used for the construction of a cheap, portable field microscope to detect malaria, that is to be done at the University of Nairobi. In a study by Isikman et al. [22] it has been shown that the malaria parasite is visible in both intensity and phase images of an infected blood cell, but perhaps more clearly in the phase image (figure 12).

6.3.2 Method

The problem of phase reconstruction is that for every pixel in the object plane, both the phase and the amplitude are unknown while the known values are half as many, since the camera only gives intensity information. The solution space of the problem is underdetermined.

In [31] a method is presented where the phase is retrieved by propagating the light back and forth between the real object plane and the virtual object plane. The phase will be updated for every iteration, while the amplitude is constrained by the intensity from the initial reconstruction of the hologram. To overcome the undetermined solution space, the object support will be determined by setting an intensity threshold. The number of
unknowns is reduced to $2 \times \text{number of pixels in object support}$. However, this method is only valid where the object support is small enough and not suitable for dense samples such as blood smear.

To be able to reconstruct object phase and amplitude for the whole sample it is possible to take holograms at different distances $z_{2a}$ and $z_{2b}$ from the camera, where in [9] $z_{2b} - z_{2a} \sim 30 - 70 \, \mu m$. The object wave should now be propagated back and forth between the $z_{2a}$ and $z_{2b}$ planes, where for each iteration the phase will be updated and the amplitude constrained. After the iteration process is finished (1–70 iterations [9]), the object phase and amplitude can be reconstructed from the intensity hologram, where the numerically determined phase has been added. Adding another plane $z_{2c}$ to the iterative process will increase the rate of convergence, however, using more than three planes will not increase the results significantly according to the study in [9].

**Registration** The recording of holograms at different heights is likely to introduce some distortion in the aligning of the sample. To accurately restore phase information, image registration of one of the holograms with respect to the other is extremely important. We have used automated feature registration in Matlab’s Computer Vision System Toolbox to find the scaling, rotation and translation of an image. Both of the holograms were first reconstructed at their respective recording distances (which are found automatically by the autofocus algorithm described in Section 6.4), then the scaling, rotation and translation of the reconstruction were found. These geometric transformations were then inversely applied to the hologram. The results of aligning with this algorithm is presented in figure 13.
Figure 13: Results when two reconstructed images of blood smear are aligned with Matlab’s Computer Vision System Toolbox. A slight rotation were introduced in the recording of the holograms and the different heights \( z_2 \) means that a small scaling is introduced. Note: there is one green and one purple image, where they overlap the result is gray.

### 6.4 Auto-focusing

To accurately reconstruct a hologram the distance \( z_2 \) between object and camera needs to be precisely determined. In this project an autofocus algorithm have been used to effectively find \( z_2 \) and minimize user interaction. As input the autofocus algorithm requires an approximate, few mm wide, interval containing the correct distance to the object plane. For the on-sensor approach, this initial guess is easily determined by the experimental set-up where typically \( z_2 \approx 0.7 - 3 \) mm.

To fine tune \( z_2 \) to a precision of 10 – 20 \( \mu m \) a focus measure \( F \) of a given image \( I \) is introduced as a variance of Sobel gradient magnitudes (equation (23)). Maximizing the variance of the Sobel gradient magnitude is a well-known technique to find the most contrast and sharpness of an image.

\[
F(I) = \frac{1}{NM} \sum_{m} \sum_{n} (\nabla S(m,n) - \bar{\nabla S})^2
\]

The input image \( I \) has \( M \times N \) pixels and \( \bar{\nabla S} \) is the mean of \( \nabla S(m,n) \). The gradient magnitudes for each \( I(m,n) \) are calculated from the gradients in the horizontal and vertical directions, \( \nabla S_x \) and \( \nabla S_y \), by using:

\[
\nabla S(m,n) = \sqrt{\nabla S_x(m,n)^2 + \nabla S_y(m,n)^2}
\]

\( \nabla S_x \) and \( \nabla S_y \) are obtained by convolving the input image \( I \) with the Sobel operators.
\[
S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}
\] (25)

The implementation of the autofocus algorithm can be found as Matlab code in the Appendix. We have used the same algorithm as is used by the Ozcan group in [30]. It should be mentioned that the algorithm fails to find the correct reconstruction distance when the background noise is too prominent or when the hologram is not good enough to give any real sharp reconstruction.
7 Results

7.1 Simulation

The key to simulating holograms is to propagate light from an object to an imaginative screen some distance $L$ away. The propagation of light can be described by the Fresnel-Kirchoff formula from Section 2. Since this is a convolution, with the convolution kernel $h$ described in 4, simulations can effectively take advantage of the convolution theorem and the fast fourier transform (FFT). To further improve efficiency it is very important to use vectorization in Matlab as described in 7.1.2.

Since simulating holograms is just the same as reconstructing holograms, just that the light is now forward-propagated instead of backward-propagated, simulations have been carried out to investigate the difference between using a FFT of the kernel $h$ (as in the Huygens convolution method for reconstruction) and the analytical optical transform function $H$ (as in the Angular spectrum method for reconstruction).

7.1.1 $H$ and $\text{fft2}(h)$

We studied the effect of using either the Huygens convolution method or the angular spectrum method by comparing the resulting kernels. The Huygens method constructs the space convolution kernel $h$ and Fourier transforms it to frequency domain where it is used for deconvolving the hologram. The Angular method instead constructs the kernel $H_A$ directly in frequency domain for direct use in the deconvolution. With knowledge of the relationship between $H$ and $h$

$$\Phi(h) = H$$

(26)

it should just be to Fourier transform $h$ to get $H$. But this is only true in the continuous case, not in the discrete. This is shown in figure 15 where two $h$ were calculated at different distances $Z$ and both where transformed using the $H_T = \text{fft2}(h)$ function in Matlab. The black areas around the central part of the $H_T$ should not actually be black but instead continue with the same pattern as in the central square. Figure 14 compares the $H_T$ to $H_A$ and shows that the middle square with information is correct but that the black area corresponds to some sort of data loss. This data loss is caused by the limited dimensions of $h$, the same dimensions as for the camera sensor used. The larger $Z$ the larger the viewed size of $h$ and tighter the crop of $h$ becomes. The frequency data of $h$ changes with the radius, i.e. higher frequencies are represented at a larger radius. A tighter crop thus removes high frequency data because it is outside of the sampling area. This limits the maximum distance at which the $H_T$ will contain enough data for providing a good reconstruction. $H_T$ will as well have problems at too small $Z$ distances due to aliasing in $h$. 

25
Figure 14: The difference between $H_T$ top and analytical $H_A$ bottom half, both show their real parts

(a) $h$, $Z = 1$ mm  
(b) $h$, $Z = 20$ mm  
(c) $h$, $Z = 100$ mm

(d) $H_T$, $Z = 1$ mm  
(e) $H_T$, $Z = 20$ mm  
(f) $H_T$, $Z = 100$ mm

Figure 15: $h$ and $H_T$ at larger distances $Z$. Columns from left to right, aliasing limit, just entering the data limit region and large data loss.
7.1.2 Vectorization in Matlab

Vectorization is a programming optimization approach in Matlab which holds opportunities for large speed improvements in comparison to the use of for-loops. It makes heavy use of the meshgrid function (ndgrid if the number of dimensions is more than three). The meshgrid takes two (or three) \((1 \times n)\)-vectors as input and organizes them in \(n \times n\) coordinate pairs, which can then be used in function calls to perform element-wise calculations. Each element in the matrix then corresponds to one calculation in a for-loop. Matlab is optimized for vector and matrix calculations and the time reduction from using this approach in our code instead of for-loops is about five times for the generation of the optical transfer function \(H\). The execution time for the for-loop was 1.64 s and for the vectorized method 0.33 s, average from four runs with an 1944 \(\times\) 1944 pixel hologram reconstruction.

7.2 Blood smear samples with the object-magnification approach

A key to improving resolution in the object-magnification DIHM approach is to decrease the size of the pinhole. However, small pinholes are hard to work with since they clog easily and requires a well-focused laser to let through any light at all. A set-up involving a pinhole and a well-focused laser also requires micrometer screws for correct aligning of the focused beam. We found such a set-up too sensitive to be used for a portable field microscope and decided to work with unfocused lasers.

We found the minimum possible diameter of the pinhole to let through enough light with an unfocused laser to be 5 \(\mu m\) (1 and 2 \(\mu m\) were too small). The numerical aperture of this quite big pinhole is 0.15, which would allow a theoretical resolution of 2 \(\mu m\) according to equation (10). Figure 16 shows the result when a screen of numerical aperture 0.073 was used. Clusters of blood cells are well resolved, but individual cells can not be clearly distinguished.

7.3 Phase recovery and autofocus results

The phase recovery method allows the reconstruction of intensity as well as phase of the object of interest. In section 6.3 it was stated that phase recovery will have particularly good effects on dense samples where the twin image can cause problems for the on-sensor approach. In figure 17 results on a blood sample are presented. The FOV of the images are zoomed in and shows the sample just when the blood cell layer starts to get more dense. In the phase image, the blood cell clusters stands out more from the background.

Figure 17 also proves that the algorithm we implemented for phase-recovery works out in practice, even though we still cannot resolve the single blood cells in the blood cell clusters due to limited quality of the camera. This will be important in future work at University of Nairobi, once a camera with smaller pixel size has been purchased.

However, the method has some complications. The autofocus algorithm is of great importance in the phase-recovery method, to be able to correctly determine the distances.
Figure 16: Left: Reconstructed hologram of blood cell clusters. The numerical aperture of the screen is 0.073. Theoretically it should be possible to attain a resolution of 4 \( \mu m \) and clusters of red blood cells are clearly visible. Right: Microscope image of red blood cells (20× magnification). \( \lambda = 635nm, d_p = 5 \mu m \).

between the two different \( z_2 \)-planes. The autofocus algorithm requires high quality holograms, so that the reconstruction of the recording distance will actually be a sharp image. Hence, the autofocus algorithm does not work for our low-resolution images.

Figure 18 aims to display some of the subtleties of the autofocus algorithm. The autofocus algorithm has been applied to different patches of the same two super resolved holograms in the phase recovery method. When used on a 3-GB-RAM laptop the super resolved holograms should be cropped to 1024x1024 pixels for reasonable execution time (about 30-60 s). For the patch to the left, the autofocus algorithm focuses both reconstructions similarly and these two reconstruction distances yield the phase image in figure 17c. Note that the difference in recording distance is 140 \( \mu m \). The patch in the middle is interesting because of two reasons. First, only the lower image is well-focused. The algorithm fails for the upper image, which is supposed to be closer to the screen than the lower image (smaller \( z_2 \)). Second, the lower image is well-focused but the recording distance is different from the patch to the left, by 110 \( \mu m \). Different patches of the sample seem to have different recording distances, which is reasonable since there might very well be instabilities in the horizontal placement of the sample in our set-up. The patch to the right shows a case when the autofocus algorithm finds the recording distance of something else than the blood smear, probably dust particles on the sample.

To conclude, the phase recovery method will require high quality holograms and preferably a precise horizontal aligning of the sample to work well. Also, when more pixels are used for the autofocus algorithm it gives better over-all results, but this is to the expense of higher computational cost.
Figure 17: Two super-resolved holograms (one of them showed in a) have been used to create a reconstructed phase image c). b) shows the reconstructed intensity image directly from the hologram and d) is the reconstruction from the hologram with its recovered phase. *Light source: LED - 435 µm, pinhole diameter: 50 µm, On-sensor approach - the FOV is cropped*
Figure 18: Results from the autofocus algorithm for different patches of the same holograms in the phase recovery method. **Left:** The algorithm finds the same focus for both SR holograms. $z_{2b} > z_{2a}$. **Middle:** Only $z_{2b}$ is well focused ($z_{2b} < z_{2a}$), but for this part of the sample the focus is 110 µm shifted as compared to d). **Right:** The algorithm finds the focus of something else than the blood smear, probably dust particles.
Figure 19: *Left:* Cropped reconstructed hologram of red blood cell clusters with LED as light source, $\lambda = 465$ nm, $d_p = 50 \, \mu m$ and $z_1 \sim 1$ mm *Right:* Reconstructed hologram of red blood cell clusters with laser as light source, $\lambda 635 = nm$, $d_p = 15 \, \mu m$ and $z_1 \sim 1$ mm.

### 7.4 Comparison of Laser and LED

Figure [19] shows the reconstruction of holograms captured with the on-sensor approach with both LED and laser. 20 holograms were used to create super resolution holograms, with the normalized convolution method, for both reconstructions. No apparent difference can be seen between the two reconstructions except for the exposure where the laser is somewhat brighter. However, the laser diode requires a smaller pinhole to attain a wide illumination cone than the LED. The laser should be used with a pinhole of diameter $10 – 20 \, \mu m$. If a pinhole diameter of $50 \, \mu m$ is preferred, it will not be possible to place the pinhole far away enough to illuminate the whole camera screen with the reference beam of the laser.

### 7.5 On-sensor resolution

The analogue resolution of a hologram depends on several different parameters in the set-up. The two approaches described in this paper differ in how the parameters effect the resolution and this section is only about the on-sensor DIHM approach parameters.

#### 7.5.1 Distance Z2

The distance between camera and object, $z_2$, is the most crucial parameter for this set-up. We found that the relationship between analogue resolution and $z_2$ was an inverse function where a closer distance meant a higher resolution. Figure [20] illustrates this by examining the first order fringe size of three objects at four $z_2$ distances, $1 – 26$ mm. $z_1$,
the distance between camera and pinhole were constant at 70 mm. The purple and green circles effectively shows why a shorter $z_2$ distance yields higher resolution. The circles are easily distinguishable from each other in figure 20a. Any problem detecting the fringes in this case is mostly due to the coarse discretization of the camera sensor used, this can to some extent be improved by super resolution.

With a larger $z_2$ the hologram fringes become less distinguishable. The purple and green ring touches each other in figure 20b and are less defined due to interference with neighbouring objects. The trend continues to the last picture at the biggest $z_2$ distance. The fringe marked with the green circle is very hard to distinguish, so hard that it would be hard to detect without the knowledge of the trend and anticipation of where it should be visible. Also note that the blue fringe now almost reaches the two others and compare that with the first image where they were separated by a large distance.

To conclude, object fringe size grows with distance, smaller fringes equals higher analogue resolution and thus shorter $z_2$ is better. This confirms the findings of the on-sensor
research team The Ozcan Group and their results with samples as close as possible to the camera.

7.5.2 Wavelength

All experiments in this report where performed with one out of the two following light sources. A red $\lambda = 635$ nm laser diode and a blue $\lambda = 435$ nm LED. A comparison of the effect on the hologram by the wavelength was done by acquiring two super resolution holograms with the exact same sample and distances only interchanging the light source. The difference is shown in figure 21. The red laser hologram 21a has a bit bigger airy disc than the blue LED 21b. This is expected from the airy disc size approximation

$$\sin \theta \approx \frac{1.22 \lambda}{d}$$  \hspace{1cm} (27)

which says that a larger $\lambda$ corresponds to a larger $\theta$. The holograms also matches simulated images of the same settings. Our conclusion from this is that a smaller wavelength does improve resolution for the same reason as a shorter $z_2$ distance does, it produces smaller fringes. The results are not overwhelming though and the best color with aspect to the object might be of higher importance.

7.5.3 Pinhole size

We did not see any effect from pinhole size on on-sensor DIHM resolution. A smaller pinhole produces a light source with a higher coherence but this is not very necessary because of the short distance from the object to the camera even a short coherence will produce the holographic patterns. Our conclusion is that the effect from pinhole size on resolution is negligible and that pinhole size should be chosen after other parameters such as, camera (same as object) size, maximum lighting distance and light intensity requirements.

7.6 Super resolution results

The performance of the normalized convolution and the cost minimization methods where compared by a small simulation where a sinc function where reconstructed from three slightly displaced under sampled images 22. 22c shows the result from normalized convolution and 22d the result from the cost minimization with an error of $10^{-5}$ It is clear that the cost minimization in this case outperforms the normalized convolution method. We chose to apply the cost minimization method as the sinc function closely resembles the airy function.
Figure 21: Comparison of hologram airy discs at different wavelengths.
Figure 22: Simulated data and results for normalized convolution and cost minimization. Up sampled four times from three displaced 64 × 64 grid.
7.6.1 The smoothing term in cost minimization

The cost minimization has two input parameters, tolerance and a smoothing term, often called lambda. The lambda, not to be confused with the wavelength in the reconstruction part of this report, acts as a smoothing term and penalizes high frequency features, see 6.2 for implementation. The effect of different lambdas is shown in figure 23. A small lambda produces a noisy output with rectangular artefacts, probably as an effect of the square big pixels in the camera and their observation model. A normal lambda used in other reports is $\lambda = 1$, figure 23c, which gives a good result but under some circumstances also produces some rectangular artefacts. A large lambda does not have any artefacts but produces a much softer HR image with some loss of valuable information in the hologram. This experiment shows that the lambda worked as one would expect and that a lambda of values other than approximately one will degrade the result either by introducing artefacts or blurring the resulting image too much. We used $\lambda = 1.2$ for our holograms, depending on how pronounced the produced rectangular artefacts were.
Figure 23: Results using different lambdas in the cost minimizations forcing function. Up sampled four times.
8 Conclusion

To detect malaria a microscope field of view (FOV) of 1.8 times 1.8 mm$^2$ is needed to be sure that enough blood cells are considered. Blood cells are 8 $\mu$m in diameter and the malaria parasite is $1 - 2 \mu$m. Digital in-line holography is a potentially cheap method of microscopy that also offers a significantly bigger FOV than conventional microscopy.

In this project a 1200 × 1600 color camera with pixel size 2.8 $\mu$m (effective pixel size 5.6 $\mu$m) has been used to record holograms with two different light sources, an unfocused laser diode of wavelength 635 nm and a LED with mean wavelength 435 nm. A pixel super resolution algorithm have been implemented to improve hologram quality. Also a phase recovery algorithm with the ability automatically find the recording distance $z_2$ has been implemented. The difference between the on-sensor approach and the object-magnification approach has been studied.

Clusters of blood cells have been successfully imaged. The achieved resolution has not been exactly determined, due to lack of a suitable test target of known size. For future work a 1951 USAF test target could be purchased.

The resolution of the object-magnification approach is highly dependent on use of a small pinhole (preferably 1 − 2 $\mu$m or less). With the use of an unfocused laser the diameter of the pinhole has been found to be limited to 5 $\mu$m to let through enough light. Focusing the laser onto a smaller pinhole would require micrometer screws and introduce too much sensitivity to aligning to be of practical use for a field microscopy.

The on-sensor approach is mostly limited by pixel size. The super resolution algorithm has been found to improve the results. The FOV of this approach is as big as the camera screen. For the camera used in this project the screen is $3.4 \times 4.8 mm^2$, which is enough for applying it to malaria detection. However, reducing the size of the pixels is of extreme importance to image smaller details. For future studies a gray-scale camera with minimum affordable pixel size should be purchased.

Mixing the two approaches (e.g. object far from pinhole, recording distance 1 − 2 cm to attain some object magnification) has proven to not work out in practice.
9 Further research

The purpose is to make an automatic setup that minimizes human interaction other than placing the sample and reading the result. To achieve this many adjustments and additional features have to be added. A software that regulates the camera has to be implemented. This software should cooperate with two stepper motors to take pictures as soon as the sample or sample and sensor have moved and also adjust the frame rate and exposure time of the camera. One of the stepper motors move the camera and sample in the $x$– direction while the second moves the sample in the $z$ direction. A conceptual model of a future possible set-up is seen in Figure 24.

With the method used now for obtaining holograms, on sensor DIHM, image processing is an essential part to achieve good resolution of the reconstructed images. An interesting refinement method called deconvolution aims to enhance the $NA$ by deconvoluting the hologram with a function that approximates the pixel function of the sensor that is used [19]. This method would further improve resolution.

An algorithm which can identify malaria infections in red blood cells has to be implemented. With pattern recognition methods and access to images with infected red blood
cell this should not be a problem. This area is already being investigated by many researchers when using microscopy images and should be realizable to implement on reconstructed holograms as well [38] [23].

The implemented algorithms are highly dependent on computational power and there are always ways to improve efficiency. Since the same computational procedure will be made for all recorded holograms a hardware designed for this single purpose can be implemented. A conventional computer has to load the holograms from the memory to the RAM and then send them to the processor. This process can be drastically speeded up by a field programmable gate array (FPGA) hardware that is solely made for running a specific series of code. This solution has been implemented for reconstruction of holograms in a PhD thesis from Lund University [25]. Also a LCD can be connected to the hardware to display results of the reconstruction and malaria identification algorithm.

It has previously been stated that the lab set-up should be cheap to enable hospitals in developing countries to be able to afford the device. A cost evaluation has been done and is seen in table 1. In comparison to the microscope used to compare the reconstruction of our holograms, which costs about 13500USD, the cost is minimal [6].

<table>
<thead>
<tr>
<th>Component</th>
<th>Price in USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Stepper motors</td>
<td>20</td>
</tr>
<tr>
<td>Pinhole</td>
<td>55</td>
</tr>
<tr>
<td>Camera</td>
<td>300</td>
</tr>
<tr>
<td>LCD</td>
<td>10</td>
</tr>
<tr>
<td>FPGA</td>
<td>90</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>30</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td><strong>~ 500 USD</strong></td>
</tr>
</tbody>
</table>

Table 1: Approximate cost evaluation in USD (May 2014) for future malaria identification set-up [30] [11] [34] [17] [11] [4] [4] [3] [3].

There will be some additional costs (stated as miscellaneous) in table 1. For instance the cost of 3D printing a holder for the set-up, screws for mounting the camera, wires for the LCD and buying a USB-cable.
10 Appendix

MATLAB CODE

%----------------------MAIN_COST----------------------

close all
%clear all
tic

% Super resolution by cost minimization

% Written by:
% Jakob Andren
% Carl Christian Kirchmann
% Elin Lundin
% Uppsala Universitet, University of Nairobi
% 2014

% Construct super resolution images from a sequence of low resolution
% images. The low resolution images must have a small
displacement shift.
%
% The low resolution images should be placed in a folder over
% the one
% with the script and relative folderpath is provided as the
% parameter
% folder

% This program uses the DIPimage toolbox and Image processing
toolbox.
% For windows run on every start of Matlab for starting DIPimage:
% run(’C:\Program Files\DIPimage 2.6\dipstart.m’)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INPUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading input
folder = ’LASER/’;
imformat = ’.bmp’;
channel = 1; % Channel Grayscale = 0, Red = 1, Green = 2, Blue = 3;
ifgauss = 0; % if 1: lowpassfilter reference beam
sigma = 100; % Lower number = higher cut frequency
% Superresolution input
Mag = 4; % side magnification of low res images (pixels * Mag^2)
lambda = 1; % Smoothing term, standard = 1, higher = smoother
tol = 1e-5; % cut interactive loop if error is lower
maxiter = 30; % absolute maximum number of iterations performed

% Image registration
[LR, px, py] = regimages(folder, imformat, channel, Mag);

% Crop LR images here if not using whole image area
LR = LR(101:400,201:600,:);

% Apply high pass filter to LR images (filter reference beam)
if ifgauss == 1
    LR = highpass_filter(LR, sigma);
end

% Superresolution
[HR, i, err] = sr_cost(LR, px, py, Mag, lambda, tol, maxiter);

% save resulting superres image, normalize to LR images
mn = min(min(LR(:,:,1)));
mx = max(max(LR(:,:,1)));

imwrite((LR(:,:,1)-mn)/(mx), 'LR_gauss.png', 'png');
imwrite(((HR-mn)/(mx), ['HR' num2str(Mag) '_' num2str(i) '.png']);

%-------------------REGIMAGES-------------------

function [LR, dx, dy] = regimages(folder, imformat, channel, Mag)

% regimages – Registration of images, function for the DIHM project

% Input:
% folder – string with relative path to folder with the LR image sequence
% imformat – ‘.png’, ‘.bmp’ Check ”help imread” for more formats
% channel – Input channel used, see loadimage for specs
% Mag – Integer up sampling constant
%
% Output
% LR – 3D dimesional matrix with loaded images stacked along the 3rd
% dimension, LR(X, Y, i) X, Y are pixelposition, i is image number
% dx – Image x displacement, dx(i) corrsponds to LR(:, :, i)
% dy – Image y displacement
% N – Calculated Superresolution grid dimensions

% This function uses the DIPimage toolbox
% On windows run on every start of Matlab:
% run(’C:\Program Files\DIPImage 2.6\dipstart.m’)

D = dir([folder ‘*’ imformat]); % load image data for later use

im1 = imread([folder D(1).name]); % Used as position reference picture
im1 = load_channel(im1, channel);

n = size(im1); % Low res size
N = n*Mag; % High res size

px_shift = 512; %pixels to be used in finding shift, even number, standard:512
minsize = min(n(1),n(2));
if minsize < px_shift % If image is smaller use whole image
    px_shift = floor(minsize/2)*2;
end
cut_x = floor(n(2)/2)-px_shift/2+1:floor(n(2))/2+px_shift/2;
cut_y = floor(n(1)/2)-px_shift/2+1:floor(n(1))/2+px_shift/2;
im_1 = im1(cut_y,cut_x); % Cropped and ready for use

% Pre allocate vectors and matrix
LR = zeros(n(1),n(2),numel(D));
px = zeros(1, numel(D));
py = px;
LR(:, :, 1) = im1;

for i = 2:numel(D)
    % Load new image on each run for saving memory
ima = imread([folder D(i).name]);
LR(:, :, i) = load_channel(ima, channel);

% Calculate the relative position
shiftvector = findshift(im_1, LR(cut_y, cut_x, i), 'iter'); % On windows run('C:\Program Files\DIPImage 2.6\dipstart.m')
you need to have DIPimage installed for running this function. Please see www.DIPlib.org

dx(i) = shiftvector(1); % DIPimage returns shifted x and y axis from what Matlab uses, keep in mind later
dy(i) = shiftvector(2);

end

%--------------------------------------------------SR_COST--------------------------------------------------------

function [z, i, err] = sr_cost(LR, px, py, Mag, lambda, tol, maxiter)

% Minimizes the cost function for finding the best estimate for the high resolution image in the super resolution construction.

% Input:
% LR – 3D dimensional matrix with loaded images stacked along the 3rd dimension, LR(X, Y, i) X, Y are pixelposition, i is image number
% px – Image x displacement, dx(i) corresponds to LR(:, :, i)
% py – Image y displacement
% Mag – Integer up sampling constant

% Output:
% z – high resolution image estimate
% i – number of iterations performed
% err – vector with errors at corresponding iteration step

% References:
% High Resolution Image Reconstruction From a Sequence of Rotated and Translated Frames and its Application to an Infrared Imaging System – Russell C. Hardiea, Kenneth J. Barnardb,
% John G. Bognarc, Ernest E. Armstrongc and Edward A. Watsonb
%
% Optical Engineering April 1997.
%
% Improved Resolution in Infrared Imaging Using Randomly Shifted Images
% – Cris Luengo – Delft University of Technology 1998

%Size parameters
m = length(px); %number of low resolution images
n = size(LR(:,1,:)); %size of low resolution images
N = Mag*n; %size of HR image

%Find maximum shift
dx_min = min(round(Mag*px));
dy_min = min(round(Mag*py));

% Make all coordinates positive
Dx = round(Mag*px) – dx_min;
Dy = round(Mag*py) – dy_min;

dx_m = max(round(Mag*px));
dy_m = max(round(Mag*py));
maxdelta = [dy_m–dy_min, dx_m–dx_min];

% Add the maximum shift to the output matrix dimensions
N = N + abs(maxdelta);

% Initial guess
z = (imresize(LR(:,1,:),N,'bicubic')+imresize(LR(:,m,:),N,'bicubic'))/2;
zprev = z;

% Load constants from file
[ alpha, alpha2, kernel ] = constants(Mag);

tic
for i = 1:maxiter

% Partial derivative of cost function
lrgrad = dnsamp(zprev, Mag, Dx, Dy, N, LR, kernel);
lrget = ceil((lrgrad–min(min(lrgrad)))/max(max(lrgrad)));

\}
zgrad = conv2(lrgrad, kernel, 'same') ./ conv2(lrgt, kernel, 'same') + lambda * conv2(z, alpha2, 'same');

% Calculate optimal steps size for iteration, epsilon in equation
gsmooth = conv2(zgrad, alpha, 'same');
zsmooth = conv2(z, alpha, 'same');
e = (sum(sum(zgrad .* lrggrad)) + lambda * sum(sum(gsmooth .* zsmooth))) / (sum(sum(zgrad .^ 2)) + lambda * sum(sum(gsmooth .^ 2)));

% Calculate iteration step
z = zprev - e * zgrad;

% Save data for analysis
stepsz(i) = e;
err(i) = std2(z - zprev); %std(z(:, floor(N/2)) - zprev(:, floor(N/2)));

% Break if low enough error
if err(i) < tol %std(z(:, floor(N/2)) - zprev(:, floor(N/2))) < tol
  break;
end

zprev = z;
disp([num2str(i) ' error: ' num2str(err(i)) ', time: ' num2str(toc)])

%-----------------CONSTANTS-----------------------------

function [alpha, alpha2, kernel] = constants(Mag)
% Generate kernels and constants for use in cost minimazation, mostly for
% in code space saving.

% calculate HR gradient image and the regulatization term
alpha2 = [0 0 0.0625 0 0;
0 0.1250 -0.5000 0.1250 0;
0.0625 -0.5000 1.2500 -0.5000 0.0625;
0 0.1250 -0.5000 0.1250 0;
0 0 0.0625 0 0];

% smooth the HR images to make the regulatrization parameter
alpha = [0 -0.2500 0];
kernel = ones(Mag);
kernel = kernel/sum(sum(kernel));
end

%-------------------------------------------------------
DNSAMP-------------------------------------------------------

function [dnZ] = dnsamp(z, Mag, dx, dy, N, LR, kernel)
% Downsamples z to LR according to the observation model

% Input:
% z – High resolution image estimate
% Mag – Magnification
% dx – pixel shift in x-direction
% dy
% N – size of HR image
% LR – array of low resolution images
% kernel – Observation model used in convolution

% Output:
% dnZ = "Low" resolution error gradient, they are placed in a HR grid

n = size(LR);

% Convolution as according to the observation model
a = conv2(z, kernel, 'same');

% Pre allocation
dnZ = zeros(N);

for i = 1:n(3)
    x = 1+dx(i):Mag*Mag*n(2)+dx(i)-1;
    y = 1+dy(i):Mag*Mag*n(1)+dy(i)-1;

    % "Low" resolution error gradient
    dnZ(y, x) = a(y, x) - LR(:, :, i);
end

%-------------------------------------------------------LOAD_CHANNEL-------------------------------------------------------

function [im] = load_channel(im_in, channel)
%loadimage loads image and preprocess it for use in the superres...

%function.

% channel =
% 0  - grayscale image
% 1  - raw bayer, RED
% 2  - raw bayer, GREEN
% 3  - raw bayer, BLUE
% 4  - weighted raw bayer

if channel == 0
    im = single(im_in);
    return
elseif channel == 1
    im = single(im_in(1:2:end,1:2:end));
    return
elseif channel == 2
    d = length(im_in)-1;
    diamond = zeros(2*d+1);
    for jj = d:-2:-d
        ii = (d-jj)/2+1;
        kk = (d-ABS(jj))/2;
        D{ii} = { [zeros(1,kk) im_in(ii,:) zeros(1,kk) ] };  
    end
    diamond = diamond + diag(D{ii}{1},jj);
    im = single(diamond(2:2:end,1:2:end));
    return
elseif channel == 3
    im = single(im_in(2:2:end,2:2:end));
    return
elseif channel == 4
    a = im_in;
    a = double(a);
    a(2:2:end,2:2:end) = a(2:2:end,2:2:end)/mean(mean(a(2:2:end,2:2:end)));
end
a(2:2:end,1:2:end) = a(2:2:end,1:2:end)/mean(mean(a(2:2:end,1:2:end)));  

a(1:2:end,1:2:end) = a(1:2:end,1:2:end)/mean(mean(a(1:2:end,1:2:end)));  

a(1:2:end,2:2:end) = a(1:2:end,2:2:end)/mean(mean(a(1:2:end,2:2:end)));  

return  

end  

%-----------------------------------MAIN_COST_WITH_PHASE-----------------------------------

close all  
clear all  

%  

*****************************************************************************

%  
%  Program to perform digital processing and reconstruction of  
%  a number of low resolution holograms at two different  
%  heights  
%  
%  Requirements: phase.m, autofocus.m, sr_cost.m, dnsamp.m,  
%  H_generator.m  
%  
%  load_channel.m, regimages.m, reconstruction.m, SR.m  
%  
%  dipImage library  
%  
%  run on every start of Matlab:  
%  run('C:\Program Files\DIPimage 2.6\dipstart.m')  
%  
%  Written by Elin Lundin, Jakob Andren, Carl Chrisitan  
%  Kirchmann  
%  Nairobi University, Uppsala University, spring 2014  
%  
%  Used on Matlab R2013a and R2013b  
%  
%  ************************************************************************************

%-----------------------------------Input parameters-----------------------------------
% Loading input
folder1 = 'level1/';
folder2 = 'level2/';
imformat = '.bmp';
channel = 3; % Channel Grayscale = 0, Red = 1, Green = 2, Blue = 3;

% Superresolution input
Mag = 4; % side magnification of low res images (pixels * Mag^2)
lambda = 2;
tol = 1e-5;
maxiter = 70;
LR_crop = [400 400];

% Gauss highpass filter input
if gauss = 1; % if 1: lowpassfilter reference beam
sigma = 100;

% Parameters for autofocus
interval = [0.0015 0.003];
tol2 = 10e-6;
nnn = 6;
cut_auto = [1024 1024]; % pixels to be used for autofocus
I1_name = 'rekonstruktion_auto_level1.png'; % name of image to write to file
I2_name = 'rekonstruktion_auto_level2.png';

% Reconstruction input
wavelength = 435e-9; % lasery')/LED wavelength
pixelsize = 2*2.8e-6/Mag;
cut3 = 1024; % Number of pixels for reconstruction of SR hologram
n_it = 70; % Iterations of phase reconstruction

tic
%

% Construct super resolved holograms
[SR_1, LR_1, x1] = SR(folder1, imformat, channel, Mag, lambda, tol, maxiter, if gauss, sigma, LR_crop, [−1 −1]);
[SR_2, LR_2, x1] = SR(folder2, imformat, channel, Mag, lambda, tol, maxiter, if gauss, sigma, LR_crop, x1);
save 'SR.mat' SR_1 SR_2;
imwrite(SR_1, 'SR1.png', 'png'); imwrite(SR_2, 'SR2.png', 'png');

disp('Superresolution done')

disp('Check SR holograms and then hit blank space to continue.');

load 'SR.mat';

%Crop for autofocusing
disp('Choose origin (upper left corner) of rectangle to be used for autofocusing');
disp('Hit blank space to continue');

pause

[point1 point2] = cpselect(SR_1, SR_2, 'Wait', true);
disp('Hit blank space to continue');

pause

point1_adj = cpcorr(point1, point2, SR_1, SR_2);
disp('Continuing'),

point1 = round(point1);
point2 = round(point2);

point1_adj = round(point1_adj);

save auto_points point1 point2 point1_adj;

load auto_points;

SR1_cropped = double(cropim(SR_1, point1_adj, cut_auto));
SR2_cropped = double(cropim(SR_2, point2, cut_auto));

%Autofocus

[z1, I1] = autofocus(SR1_cropped, interval, tol2, wavelength,

pixe lsize, nnn);

[z2, I2] = autofocus(SR2_cropped, interval, tol2, wavelength,

pixe lsize, nnn);

imwrite(I1, I1_name, 'png');
imwrite(I2, I2_name, 'png');
clear SR1_cropped SR2_cropped;

%Distance between planes

z = abs(z2-z1);
z1
z2

save 'z.mat' z1 z2 z I1 I2;

load 'z.mat';
% Aligning SR holograms
N = size(SR1); % Output view for aligned
[aligned1, aligned2, scale] = align_scale_rotation(I1, I2, SR1, SR2, N);
fused = imfuse(aligned1, aligned2);
fused = fused / max(max(max(aligned1)), max(max(aligned2)));
save('aligned.mat', 'aligned1 aligned2 scale');

% Phase reconstruction of hologram 1
[A] = phase(aligned2, aligned1, z, wavelength, pixelsize, n_it);
disp('Phase recovery done');

% Reconstruct real image
r = reconstruction(A, z1, wavelength, pixelsize);
save('reco.mat', 'r');

% Normalise for imwrite
r = r - min(min(r));
intensity = abs(r) / max(max(abs(r)));
phase = unwrap(angle(r));
phase = phase - min(min(phase));
phase = phase / max(max(phase));

% Write reconstructed image to file
imwrite(intensity, 'rek_intensity.png', 'png');
imwrite(phase, 'rek_phase.png');
toc

%----------- H_GENERATOR ------------------
function [H] = H_generator(n, m, L, lambda, pixelsize)
% Analytical generation of H (freq domain) x(y)CCD is
% pixelnumber
% H must be square, n = m
% $S = \frac{(x_{CCD}/CCD\_size(1)).^2 + (y_{CCD}/CCD\_size(2)).^2}{2}$;
% $H = \exp\left(-1i \times 2 \times \frac{\pi}{\lambda} \times L \times S\right)$;
% %
% $%k = 2\times\pi/\lambda$;

CCD\_size = [n m] * pixelsize; % [10 10] * 1e-3;
[xCDD, yCCD] = meshgrid(1:m, 1:n);

Ex = (xCDD-(n/2+1))/CCD\_size(1);
Ey = (yCCD-(m/2+1))/CCD\_size(2);
% $H = \exp\left(-\pi \times L / 1i \times \lambda \times \sqrt{1 - \lambda^2 \times (Ex.^2 + Ey.^2)}\right)$;

end

%-----------------------------------------------ALIGN_SCALE_ROTATE-----------------------------------------------

function [aligned1, aligned2, scaleRecovered] =
align_scale_rotation(original, distorted, toBeAligned2, toBeAligned, N)
%Function to align image toBeAligned with respect to original
% when
%toBeAligned is transformed by scaling, rotation and translation
% with
%respect to original

%Source: Matlabs Computer Vision System Toolbox for automated image
%registration
%For documentation see: http://www.mathworks.se/help/images/examples
%/find-image-rotation-and-scale-using-automated-feature-matching.html
% Used on Matlab R2013a

n1 = size(original); n2 = size(distorted);
%Crop images for aligning
px\_align = 1024; %pixels to be used in finding shift, even number, standard: 1024
minsize = min([n1 n2]);
if minsize < px\_align
    px\_align = minsize;
end
cut = 1:px\_align;
original = original(cut, cut);
distorted = distorted(cut, cut);

% Detect features in both images.
ptsOriginal = detectSURFFeatures(original);
ptsDistorted = detectSURFFeatures(distorted);

% Extract feature descriptors.
[featuresOriginal, validPtsOriginal] = extractFeatures(original, ptsOriginal);
[featuresDistorted, validPtsDistorted] = extractFeatures(distorted, ptsDistorted);

% Match features by using their descriptors.
indexPairs = matchFeatures(featuresOriginal, featuresDistorted);

% Retrieve locations of corresponding points for each image.
matchedOriginal = validPtsOriginal(indexPairs(:,1));
matchedDistorted = validPtsDistorted(indexPairs(:,2));

% Show putative point matches.
% figure;
% showMatchedFeatures(original, distorted, matchedOriginal, matchedDistorted);
% title('Putatively matched points (including outliers)');

% Find a transformation corresponding to the matching point pairs using the
% statistically robust M-estimator SAmple Consensus (MSAC) algorithm, which
% is a variant of the RANSAC algorithm. It removes outliers while computing
% the transformation matrix. You may see varying results of the transformation
% computation because of the random sampling employed by the MSAC algorithm.
[tform, inlierDistorted, inlierOriginal] =
estimateGeometricTransform(...
matchedDistorted, matchedOriginal, 'similarity');

% Display matching point pairs used in the computation of the transformation.
% figure;
% showMatchedFeatures(original, distorted, inlierOriginal, inlierDistorted);
% title('Matching points (inliers only)');
% legend('ptsOriginal', 'ptsDistorted');
Tinv = tform.invert.T;

ss = Tinv(2,1);
sc = Tinv(1,1);
scaleRecovered = sqrt(ss*ss + sc*sc)
thetaRecovered = atan2(ss,sc)*180/pi

outputView = imref2d(N);
recovered  = imwarp(toBeAligned,tform,'OutputView',outputView);

%recovered2 = imwarp(distorted,tform,'OutputView',imref2d(size(original)));
% figure; imshowpair(recovered2,original);
% title('Aligned reconstructions')

x = find(recovered(round(N(1)/2),:));
y = find(recovered(:,round(N(2)/2)));

aligned2 = recovered(y,x);
aligned1 = toBeAligned2(y,x);

% Crop output to same size and to quadratic images
m = size(aligned1);
M = min(m(1),m(2));
M = floor(M/2)*2;
aligned1 = aligned1(1:M,1:M);
aligned2 = aligned2(1:M,1:M);

%--------------------------------AUTOFOCUS----------------------------------

function [ z, I ] = autofocus( holo, interval, tol, lambda, pixelsize,n)
% Autofocus algorithm using Sobel gradients for maximizing contrast an to
% find focus for hologram reconstruction distance
% Input:    holo – hologram as matrix of doubles, MUST BE SQUARE !
%           interval = [start stop] – interval containing best contrast
%           tol – tolerance of z
%           n – number of planes in initial guess
% Output:   z – camera–object distance
%           I – reconstructed image at distance z

%Make quadratic
n1 = size(holo);
N1 = min(n1(1),n1(2));
N1 = floor(N1/2)*2;
ho1o = holo(1:N1,1:N1);

%Sobel operators
Sx = −fspecial( 'Sobel' ); Sy=fspecial( 'Sobel' );
G = fft2(holo);

%Initial search
L = linspace(interval(1),interval(2),n);
delta =100;

while delta > tol
    for i =1:n
        H = H_generator( N1, N1, L(i), lambda, pixelsize );
        %Reconstruct real image
        R = G./fftsift(H);
        r = ifft2(R);
        r = r/max(max(r));

        %Sobel gradients
        gradSx = conv2(Sx,abs(r)); gradSy = conv2(Sy,abs(r));
        gradS = sqrt(gradSx.^2 + gradSy.^2);
        mean_gradS = mean(mean(gradS));

        %Measure of contrast
        F(i) = sum(sum((gradS−mean_gradS).^2));
    end
    ind = find(F==max(F));
    z = L(ind);
    delta = L(2)−L(1);
    L = linspace(L(ind)−(delta−tol),L(ind)+(delta−tol),n);
end

disp(['Iterations in autofocusing: ', num2str(i)]);
I = abs(r);

%-----------------------------------PHASE-----------------------------------

function A = phase(I1, I2, z, lambda, pixelsize, n_it)
%Phase reconstruction algorithm for digital in-line holography

% INPUT: n, m pixel numbers in x,y direction
% I1, I2 intensity measurements of holograms
% z distance between object planes
% n_it number of iterations
% Output: A complex wave at hologram plane (corresponding to I1)

% I1 should be the hologram CLOSEST to the sensor

% Source: A. Greenbaum and A. Ozcan. Maskless imaging of dense samples using pixel super-

[n m] = size(I1);
H = H_generator(n, m, z, lambda, pixelsize);

phi = zeros(n,m); A=sqrt(I1).*exp(1i*phi);
for n = 1:n_it
A = A.*H; % forward propagation
phi = angle(A);
A = sqrt(I2).*exp(1i*phi); % complex field at object plane 2
A = A./H; % back propagation
phi = angle(A);
A = sqrt(I1).*exp(1i*phi); % complex field at object plane 1
end

%-----------------------------------SR-----------------------------------

function [HR, LR1, x1] = SR(folder, imformat, channel, Mag, lambda, tol, maxiter, ifgauss, sigma, cut, x2)
% Function to registerate low resolution images, manually choose what part
% of hologram to be used for super resolution, and to create super resolved
% holograms

% Input: folder – name of folder where LR images can be found
% imformrat – 'bmp', 'png' and so on
% channel – colour of light, see load_channel
% Mag – upsampling factor
% lambda – parameter of cost function to be sent to sr_cost
% tol – accepted error in iterations of sr_cost
% maxiter – max number of iterations in sr_cost
% ifgauss – 1 if highpass filter is to be applied
% sigma – parameter of high pass filter
% cut – size of LR image to be used for SR construction
% x2 – coordinates of origin of patch to be cropped in LR
% coordinate
% if x2 contain negative coordinate, a new coordinate will be asked to be picked by this algorithm
%Output: HR – superresolved image
% LR1 – one cropped LR image
% x1 – coordinate picked for origin of cropping

% Image registration
[LR, px, py] = regimages(folder, imformat, channel, Mag);
LR1 = (LR(:, :, 1) - min(min(LR(:, :, 1))))/max(max(LR(:, :, 1)));

% Apply high pass filter to LR images (filter reference beam)
if ifgauss == 1
    LR = highpass_filter(LR, sigma);
end

if x2(1)<0 || x2(2)<0
    % Choose patch of low resolutuion image to create SR image
    figure(1); hold off; imagesc(LR1); colormap('gray');
    disp('Choose origin of rectangle to be used for creating SR hologram')
    pause
    x1=round(ginput(1));
else
    x1 = x2;
end

% Shift patch if too close to edge
n = size(LR);
if x1(1)>n(2)-cut(2)
    x1(1) = n(2)-cut(2);
end
if x1(2)>n(1)-cut(1)
x1(2) = n(1) – cut(1);
end

%Crop
LR_cropped = zeros(cut(1), cut(2), n(3));
for i = 1:n(3)
    LR_cropped(:, :, i) = double(cropim(LR(:, :, i), x1, [cut(2), cut(1)]));
end

%Superresolution
[HR, i, err] = sr_cost(LR_cropped, px, py, Mag, lambda, tol, maxiter);
HR = (HR – min(min(HR)));
HR = HR / max(max(HR));
LR1 = LR_cropped(:, :, 1);
References


Populärvetenskaplig sammanfattning

I det här projektet har målet varit att undersöka en billigare och enklare metod för att identifiera malaria i blodprover. Malaria är ett stort problem i en mängd områden i världen. Flera av dessa är fattiga och lyckas i nuläget inte tillhandahålla malariadiagnosticer i den omfattning som skulle behövas. Förutom att dyr apparatur krävs, måste även personalen som sköter denna vara välutbildad och lägga ned mycket tid på att för hand granska varje blodprov för att säkerställa om en person har malaria eller inte.


Labbupptäckningen byggdes delvis med en 3D printer. Resultataterande mikroskopibilder kunde upplösa kluster av blodceller, men varken individuella celler eller dess substrukturen, som krävs för att kunna se spår av malariaparasiten. Projektet kommer att fortsättas av en masterstudent på Nairobi universitet och vi gjorde en grundlig förstudie och en utförlig beskrivning av hur han kan gå vidare med projektet. Framför allt beskrevs värden på parametrar så som avstånd mellan pinnhål och objekt, respektive mellan objekt och kamera, i uppställningen och vilken typ av kamera som ska användas för att uppnå optimala resultat.