Chapter 2

A MATLAB-based Microscope

Milton P. Macedo

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/58532

1. Introduction

When someone intends to build a laboratorial prototype of a microscope there are two major tasks. One is in the optical side which is the selection of adequate optical and mechanical components of the optical setup taking into account budget restrictions. However the major challenge is to find the adequate overall environment that enables a easy and effective integration of the different parts of the microscope in order to arrange an efficient instrument.

This is a consequence of the great evolution in microscopy field in the last decades that naturally follows the progress in science and particularly in electronics and computer science. Currently the microscopes have little similarity with the general concept of an ancient microscope. In fact those former stand-alone microscopes that were used in biology laboratories at school have moved on to a complete instrumentation system. Different areas such as optics, mechanics, electronics and software have now to be integrated in order to get a digital image of an object. Ultimately a modern microscope is a user computer-controlled instrument.

Surely this configures an instrumentation field where it is very attractive to use MATLAB. In this chapter it is presented a practical example of MATLAB application as the fundamental tool in a three-dimensional (3D) microscopy platform. The first stage of this research project consisted on the selection of a one-dimensional (1D) array CCD/CMOS sensor and the subsequent development of the sensor readout module. Afterwards the laboratory platform has been built. Besides the sensor readout module the main components of this bench-microscope are the optical layout and computer software.

The choice of an efficient computer software is fundamental as the configuration of acquisition parameters, control of data acquisition, visualization of microscope images and data processing are to be performed by the user in a computer. The versatility of this platform is probably the most important feature in order to accommodate such different tasks as:

- Acquisition of data from stand-alone sensor readout module-implementation of a data communication channel from sensor readout module to the computer;
- Positioning control of the object stage – implementation of the interface of stage actuators to computer;
- Visualization of data in real-time in a computer display;
- Development of 3D reconstruction algorithms;
- Development of Graphical User Interfaces (GUI) for the different applications.

A brief reference has to be made to the use of the Borland C compiler as computer software development tool at an initial stage of this work. Namely when testing a different acquisition hardware using a different sensor and to test this sensor readout module. But next the integration of those different tasks enumerated above had to be accomplished. At this point MATLAB had been naturally considered owing to its versatility given by the functionalities from its core and toolboxes.

In this chapter the focus is the description of MATLAB applications. However for a deeper comprehension of MATLAB functionalities implemented in these applications some details of the bench-microscope prototype have to be stated. Then firstly some hardware features as the sensor readout module and object stage positioning are reported. The particularities in image build and visualization owing to the use of a linear image sensor in this bench-microscope are also covered.

On the other hand the best mode of evaluating the effectiveness of the MATLAB applications is showing the results obtained with this bench-microscope. Four applications have been developed for the implementation of image acquisition and visualization as well as for the assessment of image quality and image processing in some practical applications of this platform in materials science field.

Lastly a summary of the overall functionalities of these different MATLAB applications and a discussion of the advantages of a platform that use such a diversity of integrated tools is presented.

2. Microscope implementation

The challenge of assembling a microscope with the diversity of areas of knowledge that it demands had led to the development of open-source microscopy software. In this manner different research groups in universities as well as in industry work together in order to build software platforms that make easier the implementation of the different tasks in order to accomplish the acquisition of an image on a microscope.

Obviously the use of an open source microscopy software should be considered whenever a new microscope setup is assembled. Amongst these open source microscopy software, *Micro-Manager* and *OME* (*Open Microscopy Environment*) are probably the better established.

Micro-Manager is a complete microscopy software that include control of microscope itself and associated hardware. A long list of microscope models as well as cameras, stages and other peripherals can be controlled. It runs as a plugin to *ImageJ* and provides an easy to use software to control microscope experiments.

OME aims at the storage and manipulation of biological data. It comprises a client-server software (*OMERO*) for visualization, management and analysis of images and a *Java* library (*Bio-Formats*) for reading and writing biological image files. This library can be used as an *ImageJ* plugin, MATLAB toolbox or in our own software. *ImageJ* is a public domain *Java* image processing program. It is a very complete image tool that can be used with many image formats as well as raw-images.

As this is a research project a fast and easy access to hardware in order to test other acquisition and control configurations would be important. Surely if a software platform is developed from the zero it is more versatile and flexible. On the other hand the graphical user interface (GUI) was adapted from the one used in the preliminary tests which had been developed in C/C++language. These two issues together with other particularities listed in table 1 were decisive to the choice of developing in MATLAB an entirely original and dedicated software for this project.

Micro-Manager	OME	Original MATLAB applications
Disadvantages		**Advantages**
Sensor and stage actuators devices not supported	Do not perform microscope control	Versatility and flexibility in hardware configuration
Time consuming and energy investment in Micro-Manager	Optimized to work with biological data	GUI layout previously developed in C/C++ language
No previous experience in using ImageJ	No previous experience in using ImageJ	Experience in programming languages / MATLAB

Table 1. Advantages and disadvantages of using an open source microscopy software or develop original MATLAB applications for this project.

3. Description of bench-microscope

The purpose of this work was to build a platform to develop and test algorithms able to obtain 3D images. The own microscope optics is based on a linear-array image sensor. At an initial stage an hardware previously developed in our research team, named PAF (*Photodiode Array Fluorometer*), had been used. After the implementation of a few and plain adjustments in the PAF software the first tests were performed. A scheme of this system is shown in figure 1.

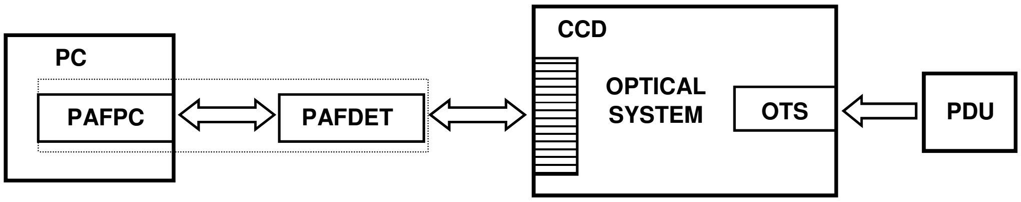


Figure 1. Test system using PAF. PAFPC e PAFDET are PAF hardware modules in computer bus and with detector, respectively; OTS – Object Translation Stage; PDU – Piezoelectric Drive Unit.

Owing to the excessive pixel width in the linear-array CCD used in PAF other sensor must be searched. A new sensor in the market (*LIS-1024 from Photon-Vision Systems*) with 1024 pixels of 7,8 μm width had been selected. Afterwards the development of a sensor readout had also enabled to assemble the sensor in the optical bench. The block diagram showing microscope architecture is presented in figure 2 as well as a photo of the optical layout showing the sensor-readout module and stage actuators both controlled from MATLAB. The optical layout is beyond the scope of this chapter but for clarity a brief description of sensor-readout module and positioning of the object platform is essential.

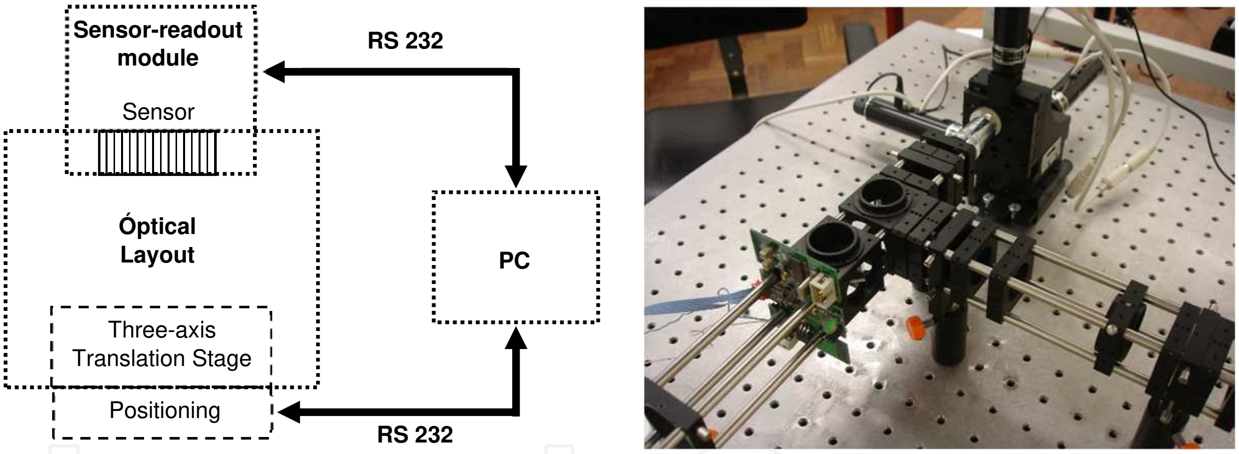


Figure 2. Block diagram and a photo of the bench-microscope.

3.1. Sensor-readout module

This stand-alone module is based on a microcontroller of PIC family *(PIC16F876)* from Microchip. It had been selected amongst a set of similar devices as it completely meets the predefined specifications, namely: a 10-bit ADC, three timers and an high versatility owing to an interrupt structure with thirteen interrupt sources.

Its weakness lies in communication options. It only has a USART for RS232 communication. So sensor data is transferred to the computer through its serial port (RS232). However the optimization of the system regarding acquisition speed is not a goal of this project. Otherwise other PIC microcontroller, *PIC16C745*, with an USB serial port would be the right choice. But the overall specifications of *PIC16F876* are more adequate to the system needs namely because of its 10-bit ADC in comparison to the 8-bit ADC in *PIC16C745*.

Figure 3 shows the block diagram of the module. Besides the sensor and microcontroller it contains a RS-232 *driver* (*MAX242 from Maxim*) that receive/transmit signals from/to PIC serial port. This driver also put data in electric format of RS-232 standard and manage control signals for data communication with the computer. As serial ports have been gradually disappearing of computers in recent years there is the possibility of using an USB to RS232 adapter and connect this module to a more modern and powerful computer.

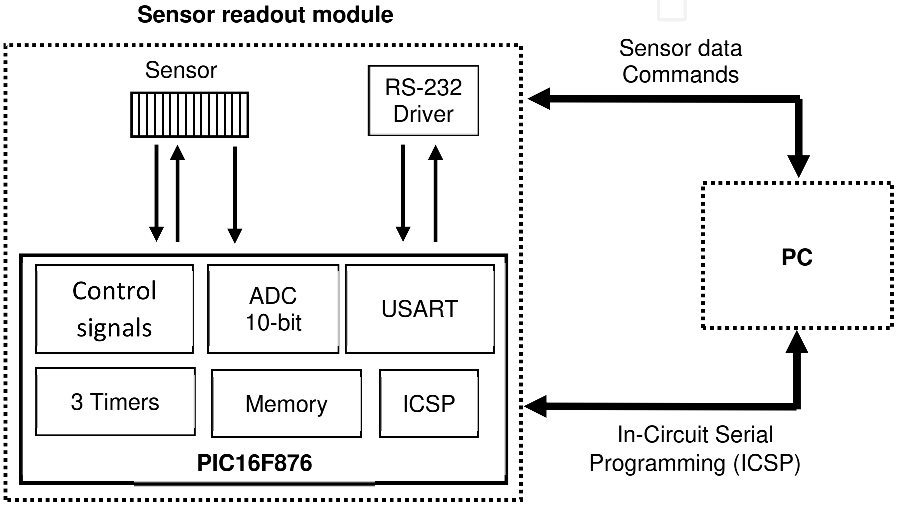


Figure 3. Block diagram of the sensor readout module.

Another important functionality of this PIC microcontroller, shown in this block diagram, is called ICSP (*In-Circuit Serial Programming*). This enables an easy mode for programming the PIC, changing its configurations, namely the sensor readout mode, without the need of take the hardware module out of optical layout.

3.1.1. Sensor readout

The ICSP functionality also enables a fast and easy configuration of the sensor operation mode taking advantage of its flexibility that comes out from CMOS technology. It is very useful as the more adequate sensor operation mode depends on the type of application in which the bench-microscope is used.

The sensor readout mode normally used was the destructive Dynamic Pixel Reset (DPR). The reset of each pixel is executed as the respective readout is concluded. This assures an equal integration time for every pixel. To ensure the correct timing of the readout start of the next set of 1024 pixels, the PIC waits by a specific control signal from the sensor.

The verification of timing specifications for sensor readout is ensured by the three *timers* of the PIC. Therefore the internal clock of the sensor with a duty-cycle of 50%, the readout of each pixel data in the second half of the respective readout time window and the acquisition time in accordance with precision specifications of the ADC are easily implemented.

The acquisition cycle is based on the appropriate interrupt structure of the PIC. Figure 4 presents the timing diagram of the acquisition cycle. It is also shown the timing of data transfer to computer through RS-232 serial port.

No external memory exists in this stand-alone module. As there is no way to store the data in the module memory, each pixel value of 10 bits, the result of the ADC conversion of the analog value read from each sensor pixel, has to be sent to computer till the end of the timeslot. In this case the timeslot is the time lapse from one ADC conversion to the next one.

These 10 bits value from each pixel is packed in a frame with three words of 8 bits (bytes) as it is shown in figure 5. Thus each timeslot must be long enough for the USART complete the transfer of this frame.

Using the *Instrument Control Toolbox* the configuration of the computer serial port was performed. Preliminary tests had shown that using the maximum baud rate of 59200 bps the communication errors were very scarce. In spite of this it had been considered that a baud rate of 19200 bps was the best compromise between speed and reliability. The option had been to completely avoid these error relaxing the speed goals.

This lower baud rate imposes a rate of pixel sensor readout slightly above 1 kHz. This is achieved from a timeslot width (T_{cycle}) of nearly 900 μs. Therefore the acquisition of all 1024 sensor pixels takes around one second.

In many applications of this bench microscope it is unnecessary to perform the acquisition of the 1024 pixels. Owing to the easiness of programming the PIC that arises from the ICSP functionality described above it is plain to change the sensor readout configuration namely the timing issues in order to adequate it to the specific needs of each situation.

In these cases instead of wasting time for the acquisition of data without relevant information it is clearly useful the definition of a region of interest (ROI). The dimension of this ROI in image plane depends on the magnification of the objective used. One of the objective lenses in this bench microscope is 40x / 0.65 NA (numerical aperture is a number related to the width of the cone of light gathered by the lens). So the image with 1024 pixels corresponds to *200 μm* in the sample (in object plane). If in one specific application this is excessive a ROI should be defined and consequently the acquisition rate is increased. For a ROI with 256 or 128 pixels the acquisition rate is increased by a factor of 4 or 8, respectively.

3.2. Positioning actuators

This bench microscope is intended to be used in reflection mode. Its optical layout is an epi-illumination configuration, typical of confocal microscopes. In this case light travelling from the light source to the sample has a fraction of its path in common to light reflected by the sample. Due to budget constraints it has been made the option by a stage-scanning instead of the beam-scanning configuration.

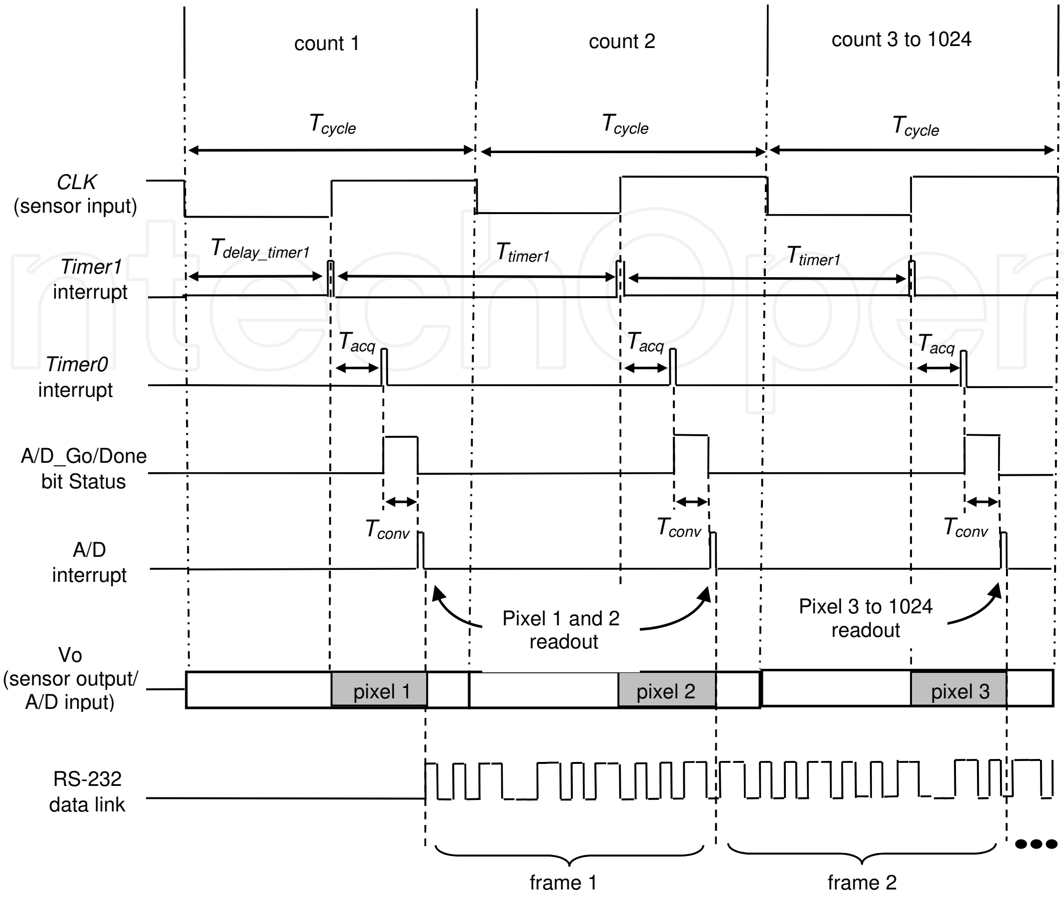


Figure 4. Timing diagram of the acquisition cycle of sensor data (time intervals not in scale). Pixel acquisition time (T_{cycle}) of 888 µs, A/D acquisition time (T_{acq}) of 30 µs and conversion time (T_{conv}) of 20 µs.

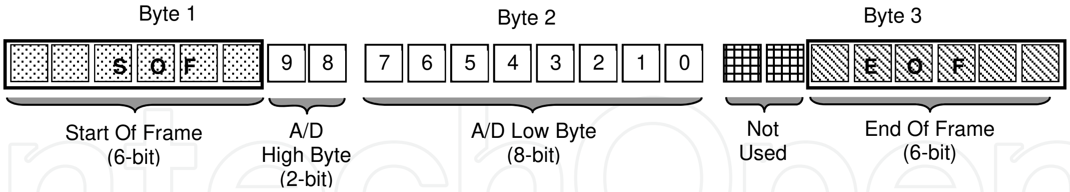


Figure 5. Format of the *frame* used in RS-232 communication.

There are a wide range of positioning devices that use, e.g., stepper motors, acousto-optic deflectors (AOD), galvanometric mirrors or piezoelectric drivers. The selection of the actuators to control the positioning of the object translation stage had been based on the following issues:

- Easiness to accommodate in the three-axis translation stage (*Melles Griot 17 AMB 003*);

- Computer-controlled;

- Cost effective (nice compromise between cost and performance).

T-series positioning products from Zaber^{TM} and in particular linear actuators are ready to mount in the translation stage. These computer controlled positioning products use stepper motors to achieve open loop position control. These devices turn by a constant angle called a step for every electrical impulse sent to them. This allows a system to be built without feedback, reducing total system cost.

However, being incremental (as opposed to absolute) in nature, the stepper motor must initially be zeroed by going to a home sensor. As there is no encoder, the actual position of the device will become different from the position shown in the computer display. Also these positioning products use a direct drive system for a simplified mechanical design with no coupling, gear, belt or other expensive components.

Likewise the specifications of this linear actuators in terms of resolution and repeatability comply with the purposes of the bench-microscope thence the option by the linear actuator T-LA 28, with a 28 mm range.

The resolution (or addressability) is the distance equivalent to the smallest incremental move the device can be instructed to make. In other words, it is the linear or rotational displacement corresponding to a single microstep of movement. The resolution for T-LA products is 0.09921875 mm (or approximately 0.1 mm).

The repeatability is the maximum deviation in the position of the device when attempting to return to a position after moving to a different position. The typical repeatability for T-LA actuators is about 0.3 mm. Also the typical backlash, which is the deviation of the final position that results from reversing the direction of approach, is 2.2 mm. T-LA devices have built in anti-backlash routines that do not affect motion in the positive direction (increasing absolute position, plunger extending). For negative motion, however, the device will overshoot the desired position by roughly 600 microsteps and return, approaching the requested position from below.

3.2.1. Positioning control

Image acquisition with this bench-microscope requires to control the positioning of the object stage. Automatic scanning in the three-axis (XYZ) is performed. One T-LA 28 actuator controlled through the RS232 port of a computer is used to implement the scanning in each axis. However the three units are connected in a daisy-chained mode thus sharing the same serial port in the computer. The configuration of the computer serial port and the control of RS232 communication was implemented in a MATLAB application.

Communications settings must be: 9600 baud, no hand shaking, no parity, one stop bit. After power-up, the units in the chain will each initialize itself as unit #1 and thus each will execute the same instructions. To assign each unit a unique identifier, a renumber instruction must be issued after all the units in the chain are powered up and every time a unit is added or removed from the chain. Instructions must not be transmitted while the chain is renumbering or the renumbering routine may be corrupted. Renumbering takes less than a second, after which instructions may start to be issued over the RS232 connection.

All instructions consist of a group of 6 bytes. They must be transmitted with less than 10 ms between each byte. If the unit has received less than 6 bytes and then a period of more than 10 ms passes, it ignores the bytes already received. The table 2 below shows the instruction format:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Unit #;	Command #	Data (least significant byte)	Data	Data	Data (most significant byte)

Table 2. Instruction format.

The first byte is the unit number in the chain. Unit number 1 is the unit closest to the computer, unit number 2 is next and so forth. If the number 0 is used, all the units in the chain will execute the accompanying command simultaneously.

The second byte is the command number. Bytes 3, 4, 5 and 6 are data in long integer, 2's complement format with the least significant byte transmitted first. How the data bytes are interpreted depends on the command.

Most instructions cause the unit to reply with a return code. It is also a group of 6 bytes. The first byte is the device #. Byte 2 is the instruction just completed or 255 if an error occurs. Bytes 3, 4, 5 and 6 are data bytes in the same format as the instruction data byte. For some instructions in this reply it is sent the actual effective position.

Therefore the communication between the computer and T-LA 28 is executed in both directions. From the graphical user interface (GUI) of the MATLAB application the user gives an order to perform a command, e.g., reset, home, renumber, move absolute, move relative, using the instruction format presented in table 1.

The slow data communication between these linear actuators and the computer, 9600 *baud*, is one of the most important weaknesses of these actuators. Image acquisition rate is consequently low but it is assumed that in this work the concern is not to optimize this rate. The aim of this project is to obtain microscope images from which a three-dimensional reconstruction is made. Therefore with this goal in mind the easiness in the integration of this actuators in the optical layout combined with programming versatility largely compensate the imposed low acquisition rate.

4. Image visualization

Raw-data provided in one sensor readout consists of 1024 analogue values that the A/D in microcontroller converts in 10-bit digital values. And these values compose a linear image of a region-of-interest (ROI) in the object with a length that depends on the objective magnification. It corresponds to 200 μm and 400 μm for 40X and 20X objectives, respectively,

Usually the object plane is considered as the XY plane. Hence the optical axis which is perpendicular to this plane is the z-axis. When using a linear sensor normally the x-axis represents the direction of the sensor pixels. Thus the acquisition of a two-dimensional (2D) image implies the scanning of the object on the translation-stage in the other lateral direction perpendicular to sensor (y-axis).

4.1. 2D image view

Two-dimensional image build is schematically shown in figure 6. This diagram illustrates its dependence on spatial sampling rate in both lateral axes.

Sampling rate in x-axis is imposed by pixel width which is 7,8 µm. It is possible to join contiguous pixels and consequently the correspondent photoelectrons are added. This process is denominated by binning. The advantages are a faster image acquisition rate, reduced exposition time or an improved signal-to-noise ratio (SNR). However it is achieved at expense of a degradation in image resolution [1]. Spatial sampling rate in object space also depends on objective magnification. For the objectives 40x and 20x these values are 195 nm and 390 nm, respectively, without binning. Consequently putting together e.g., four contiguous pixels, the sampling rate is decreased by the same factor.

Spatial sampling in y-axis is imposed by the minimum length in XYZ stage movements in that direction. Based on the specifications of the positioning devices the option was to use the minimum scanning step of 1 µm. On the other hand in microscope applications that demand a larger field-of-view (FOV), wider scanning steps, till the maximum of 20 µm, were used at expense of image lateral resolution.

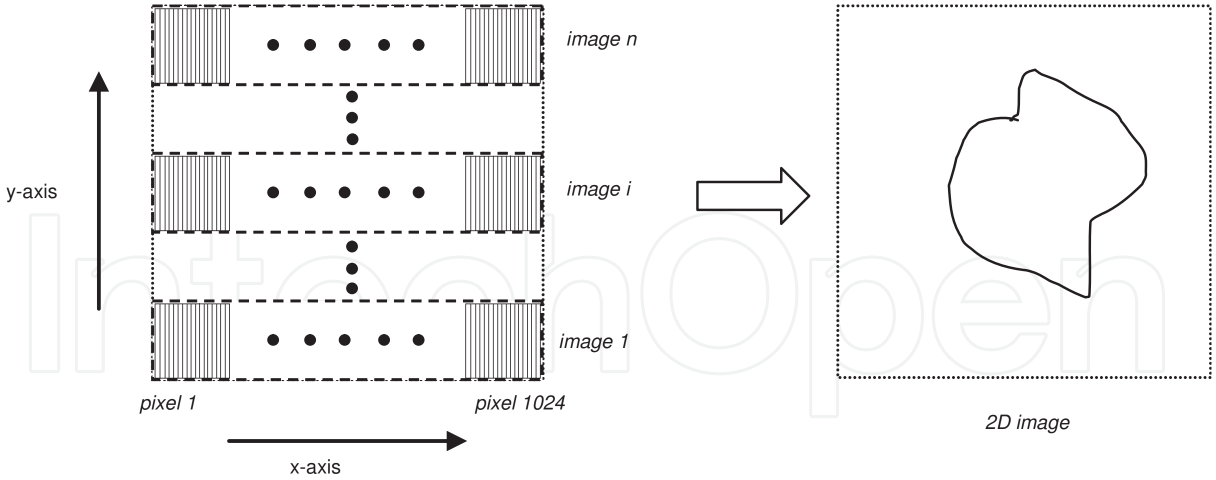


Figure 6. Schematic of 2D image build. Pixel width is 7,8 µm at image space. Sampling rate in y-axis at object space range from one to twenty microns.

To build a 2D image that is a faithful representation of the relative dimensions of the object in the two lateral directions it is necessary to have an equal scale in both axes. As spatial sampling rate in the two axes generally is different, the MATLAB function *imresize* is used. One of its parameters is exactly the ratio of sampling rate values in both axes. This parameter acts as a multiplicative factor to be used for the values in the axis with lower sampling rate. If the 20x objective is used together with the minimum scanning step in y-axis a multiplicative factor of 2,56 should be used in *imresize* function. Other parameter is the interpolation method that may be chosen from the following three: nearest-neighbor, bilinear or bicubic.

This process for 2D image build is illustrated in the scheme of figure 7. It is presented as example one image of the USAF resolution target used in this work for image quality assessment.

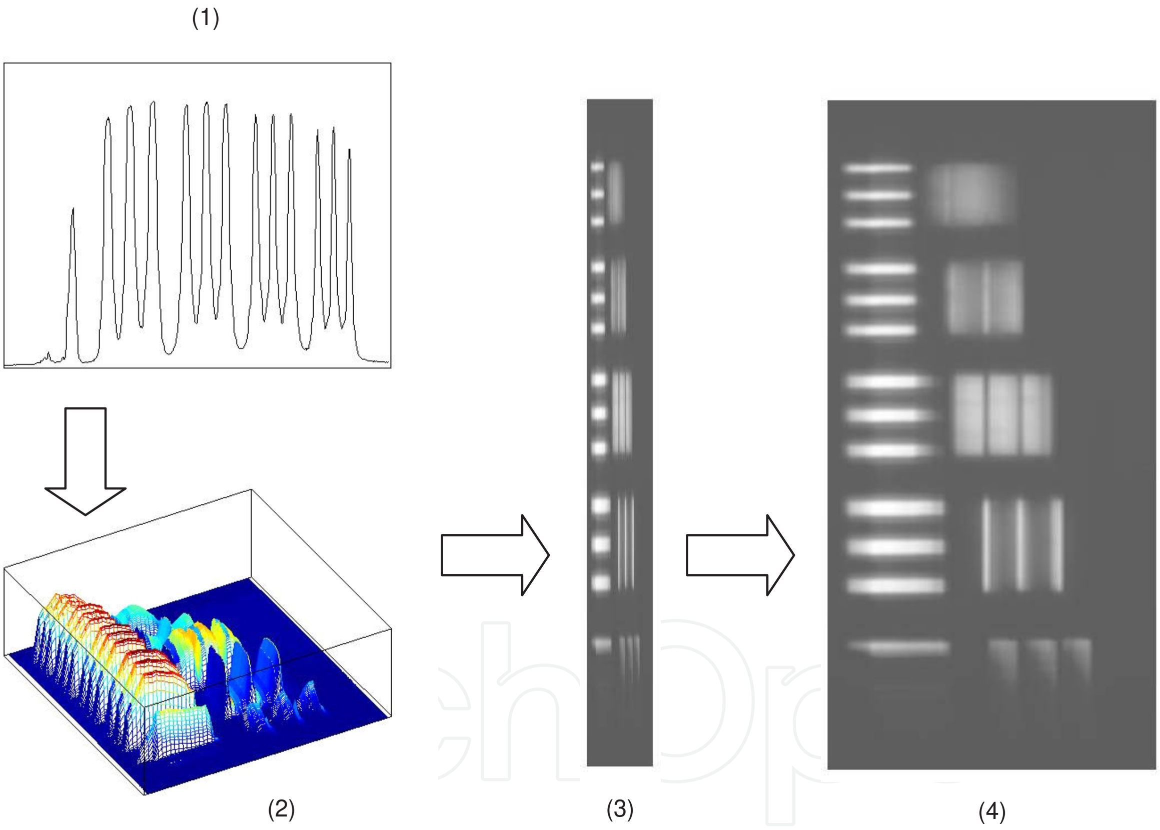


Figure 7. Diagram depicting the 2D image build in the case of the USAF resolution target. (1) Linear image (data from one sensor readout); (2) 3D representation of the intensity values in the XY plane (after the scanning along y-axis); (3) 2D image using data from sensor readout (no data processing); (4) 2D image with equal scales in both lateral axes (output of *imresize* function). In (3) and (4): horizontal direction – y-axis; vertical direction – x-axis.

In some microscope applications the field-of-view from each sensor readout is insufficient. However this microscope is able to increase the field-of-view through the scanning also along sensor direction (x-axis). The overall width of the 1024 sensor pixels gives a total field-of-view of approximately 400 μm and 200 μm, for 40x and 20x objective lenses, respectively. In order to build the 2D image with a larger field-of-view it is necessary to use an adequate method to mount contiguous images in x-axis. The MATLAB function *montage* performs this action assuring a correct position alignment. In the overall image these transitions are undistinguishable.

4.2. Extraction and visualization of 3D information

Linear images from each sensor readout of its 1024 pixels as well as 2D images obtained in result of the scanning in y-axis are blurred by light collected from reflection on points of the object at different depths. Then this blur in an ideally focused image is a result of light coming from reflection on points in planes in front or behind the focal plane.

Scanning optical microscopes (SOM) and particularly the confocal microscope, which is the most widely used, are suitable to get three-dimensional (3D) information. To achieve an image with 3D information the acquisition of different optical sections must be performed. It consists on images of object planes at different depths. Its acquisition is performed through the scanning along the direction of the optical axis, usually known as the axial direction.

Thus as this bench-microscope is intended for the acquisition of images with 3D information, scanning of the XYZ object translation stage is also performed along optical axis (z-axis). Spatial sampling rate in this axis is similar to the used in y-axis. Sampling intervals range from the minimum of 1 μm till 20 μm. The axial resolution depends on the numerical aperture (NA) of the objective. So the selection of the spatial sampling rate must have in consideration which objective is used, namely the 0.4 NA or 0.65 NA.

Amongst the different modes of visualization of 3D information the following two are the more usual:

- Auto-focus images – images with three-dimensional information as points in image are focused at different depths.
- Topographic maps – a three-dimensional representation of the height, $h\,(x_i,\ y_i)$, of each point in XY plane.

A more detailed description of the methods implemented in this microscope will be presented together with the presentation of several microscope applications. However to illustrate how 3D information can be visualized and simultaneously to evidence the ability of this bench-microscope to achieve this goal see figure 8. It is the result of the application of a particular non-automatic method. In this case the user select for each lateral position the best focus on the bonding wire amongst the overall axial positions. It is clearly distinguishable the wire inclination that departs from the pad on the integrated circuit and increases its height from the left to the right.

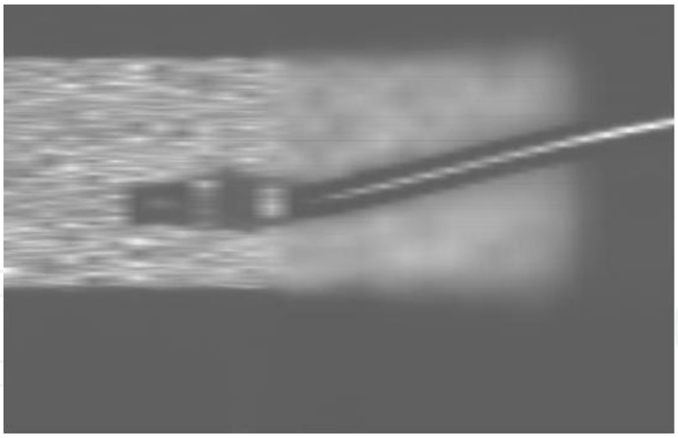


Figure 8. *Best-focus* image (bonding wire). FOV: 620 μm x 400 μm (horizontal direction – y-axis; vertical direction – x-axis).

5. Results

5.1. Image acquisition

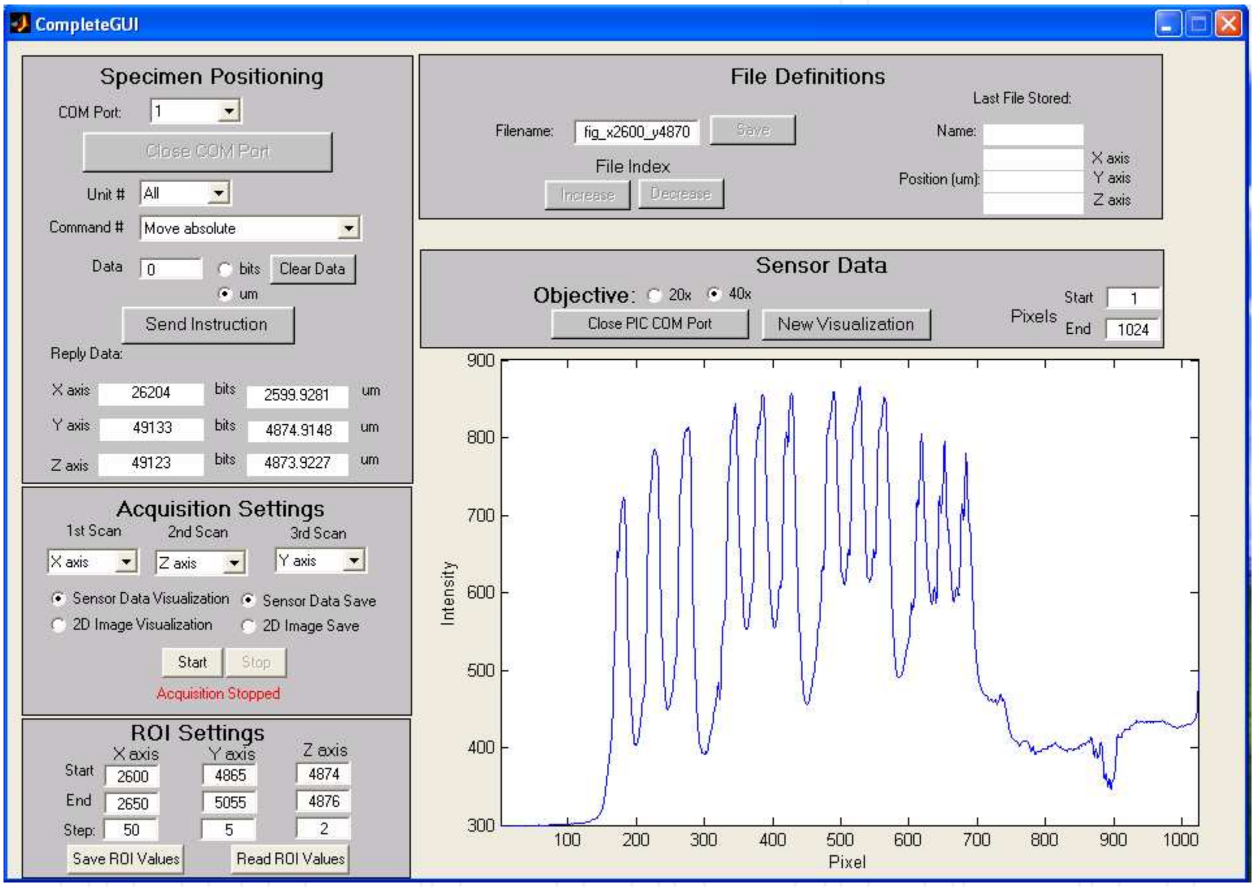
The image acquisition as well as its visualization is configured by the computer user. Even in the early stage of this bench-microscope in which preliminary tests were performed with PAF hardware and software it was accomplished. Using PAF system as well as in the initial tests using this sensor readout module the software development tool had been Borland C. This occurred mainly because the software already developed with this tool for PAF was in an easy and fast mode applied to this new hardware module.

However a point had been reached in which the need of improving the quality of the graphical user interface and of creating a more user-friendly interface had demanded the change to other software environment. Borland C had been a great tool to develop code for someone that has learned C language in the first step of programming learning. Nevertheless it was not directed for the development of graphical user interfaces (GUI). Consequently it was a time consuming task and the graphical interface had not reached the desired quality.

As already mentioned the option had been to use MATLAB. In fact it is a very complete tool as through the use of its objects it is easy and much faster to create a graphical user interface, to add or remove any functionality at any time. But in this bench-microscope there is also the necessity of implement the acquisition of sensor data as well as the control of the positioning of the object stage. Yet MATLAB covers this scope through its toolboxes namely *Instrument Control* or *Data Acquisition*. The mode how this functionalities had been easily implemented in MATLAB had been probably the major proof that it was the best choice.

CompleteGUI had been the first MATLAB application developed for this bench-microscope. It is the more general in the sense it is used anytime the user intends to get a microscope image. Besides the initialization of the sensor and positioners the user has to define the acquisition parameters in terms of the scanning axes, range and steps.

As the sensor readout is performed with an acquisition rate of about one frame of 1024 pixels per second and it is a linear image sensor it is necessary to complete the acquisition of a set of frames to build a two-dimensional (2D) image. It usually takes a few tens of seconds. This image is then built from a set of one-dimensional (1D) images which are also displayed in real-time. One functionality of this application is to build this 2D image giving the possibility of image visualization immediately after its acquisition is completed. This GUI is shown in figure 9 with an example of real-time visualization of raw-data from sensor which is the 1D image.



CompleteGUI

Specimen Positioning

COM Port: 1

Unit # All

Command # Move absolute

Data 0 ○ bits ● um

Reply Data:

X axis	26204	bits	2599.9281	um
Y axis	49133	bits	4874.9148	um
Z axis	49123	bits	4873.9227	um

File Definitions

Filename: fig_x2600_y4870

File Index

Last File Stored:

Name:

Position (um): X axis Y axis Z axis

Sensor Data

Objective: ○ 20x ● 40x

Pixels Start 1 End 1024

Acquisition Settings

1st Scan	2nd Scan	3rd Scan
X axis	Z axis	Y axis

● Sensor Data Visualization ● Sensor Data Save

○ 2D Image Visualization ○ 2D Image Save

Acquisition Stopped

ROI Settings

	X axis	Y axis	Z axis
Start	2600	4865	4874
End	2650	5055	4876
Step:	50	5	2

Figure 9. Graphical User Interface – *CompleteGUI* – developed for configuration settings and acquisition control and visualization.

The functionalities of *CompleteGUI* application are summarized below:

- Object-stage positioning
    - User definition of the order of scanning in the three axes (through the GUI);
    - User definition of the region-of-interest limits in the three axes (through the GUI);
    - Send commands to positioners through a computer serial port.

- Sensor data readout
  - Send a command to the sensor readout module to signal acquisition start (through a computer serial port).
  - Receive the sensor data, 10-bit values of the 1024 pixels (through a computer serial port).
- Image visualization
  - Real-time visualization (preview) of each sensor readout, i.e., a 1D image that shows sensor raw-data;
  - Visualization of each 2D image as soon as the acquisition of the set of sensor data is completed
- Data-files creation
  - Open Excel data-files
  - Store data files in the computer hard disk.

5.2. Image quality assessment

One of the most important tasks to be performed had been image quality assessment. After the implementation of overall system architecture that includes the optical layout it was necessary to know whether the images obtained with this bench-microscope are in good agreement with theoretical expressions. In fact the performance of a microscope is determined by the quality of its output image. There are two different approaches for this assessment:

- Quantitative assessment through the determination of the modulation transfer function (MTF) and point spread function (PSF) which are a measure of contrast and resolution, respectively.
- Qualitative assessment using images of specific objects with known dimensions [2]

Usually only one of these two different methods is selected. It depends on the specificity of the system. Owing to practical reasons a method for quantitative assessment was used. Namely because this is a bright-field microscope in reflection mode and a new method for the determination of PSF would be implemented.

5.2.1. Contrast measurement (determination of MTF)

Firstly it was measured the MTF which is an usual representation of the performance of imaging systems. It is used as a measure of image contrast. The general definition of MTF is given by the ratio of modulation depth in the output and input of an optical system in function of the spatial frequency when a sinusoidal target is used as input [3]. Modulation depth (*M*), normally called contrast, has the following definition:

$$M = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}, \tag{1}$$

where I_{max} and I_{min} are the maximum and minimum intensity values, respectively. It is relative either to irradiance emitted by the object or collected in the image, which are system output and input, respectively. Thus MTF definition is:

$$MTF(\xi) = \frac{M_{output}(\xi)}{M_{\underset{input}{\sin usoidal}}(\xi)}, \tag{2}$$

whereξ is the spatial frequency (rigorously it is its component along sinusoidal grid direction)

From the different methods for MTF determination [4], the choice had been the scanning method. It consists on the measurement of the dependence of the contrast on spatial frequency using a sinusoidal grid as object. Although the MTF concept is applied to sinusoidal grids, for practical reasons, it is much easier to use square grids.

A widely used square grid is the USAF (*United-States Air Force*) resolution target. Then MTF determination was accomplished through contrast measurement for each group / element in images of the USAF target. This target consists on groups of three lines with different densities. The separation of the lines ranges from a minimum of approximately 2 μm (that corresponds to a maximum spatial frequency of 228 lp/mm) to tens of microns (one line pair/mm).

In this manner this method consists on the measurement of the system response to an input which is a square and not a sinusoidal grid. This is the definition similar to MTF but designated by contrast transfer function (CTF), in accordance to:

$$CTF(\xi_f) = \frac{M_{output}(\xi_f)}{M_{\underset{input}{square}}(\xi_f)}, \tag{3}$$

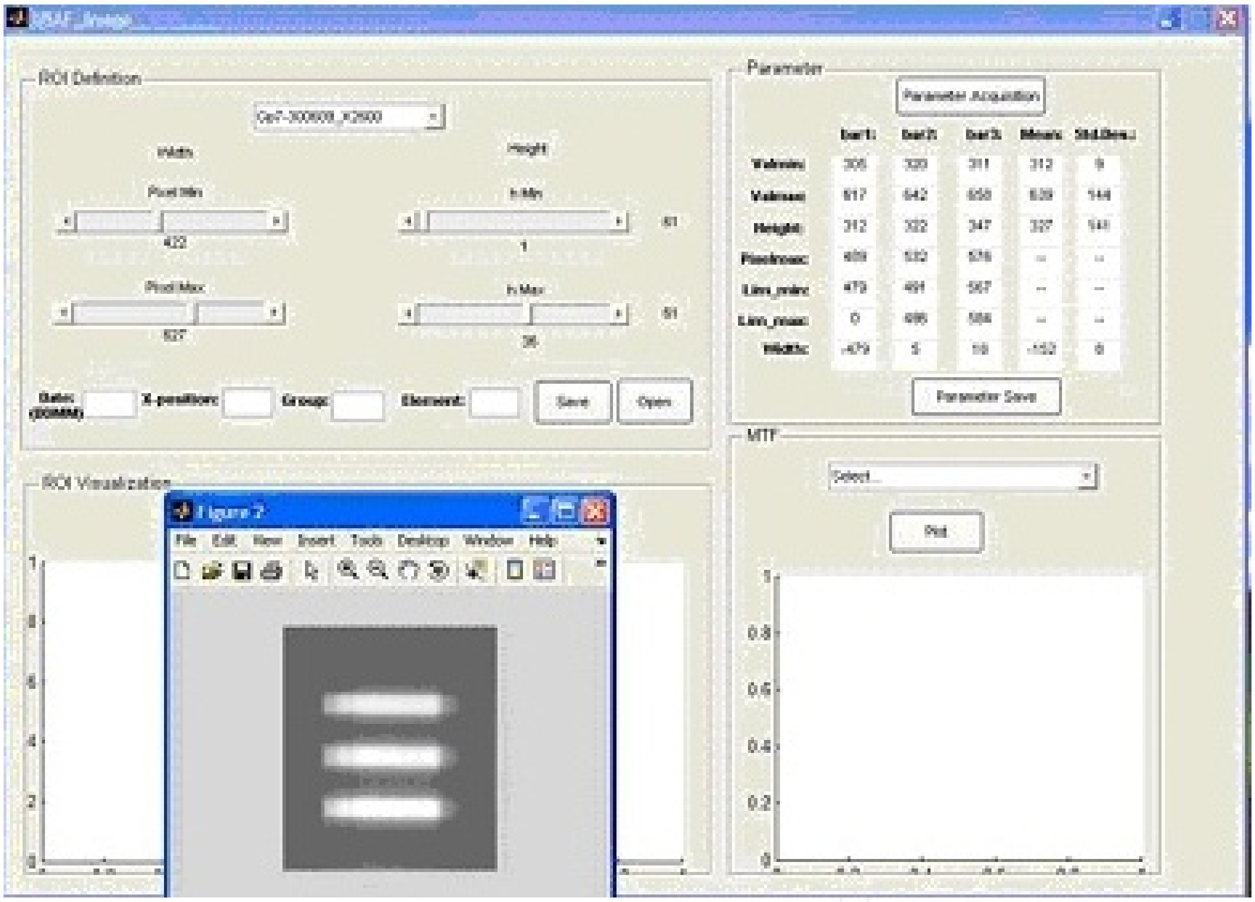
whereξ_f is the fundamental spatial frequency.

To materialize this implementation the MATLAB application *USAF_image* had been developed. Its graphical user interface is presented in figure 10.

The functionalities of *USAF_image* application are summarized below:

- Region-of-interest (ROI)
  - User definition of ROI limits in the two axes (through the GUI);
  - ROI visualization;

- Image files
  - Store image files in JPEG format in the computer hard disk;
  - Parameter acquisition
- Selection of a ROI containing one line;
  - Determination of maximum and minimum value in that ROI;
  - Determination of line width given in number of pixels using a predetermined threshold value (usually average of maximum and minimum values).



	bar1:	bar2:	bar3:	Mean:	Std.Dev.:
Valmin:	306	320	311	312	9
Valmax:	617	642	658	639	144
Height:	312	322	347	327	141
Pixelmax:	409	532	576	--	--
Lim_min:	479	491	567	--	--
Lim_max:	0	496	584	--	--
Width:	-479	5	18	-152	8

Figure 10. Graphical User Interface – *USAF_image* – developed for image quality assessment namely contrast measurements.

CTF values were then calculated for each spatial frequency, using the groups of three lines in USAF target images as shown in figure 11. Different optical configurations had been used in the acquisition of these two images. Images on the left and right correspond to wide-field and line-illumination, respectively. Despite the importance of the illumination mode for the success of this bench-microscope, it is out of the scope of this chapter. Just for completeness the comparison of image contrast on these two illumination modes will be presented.

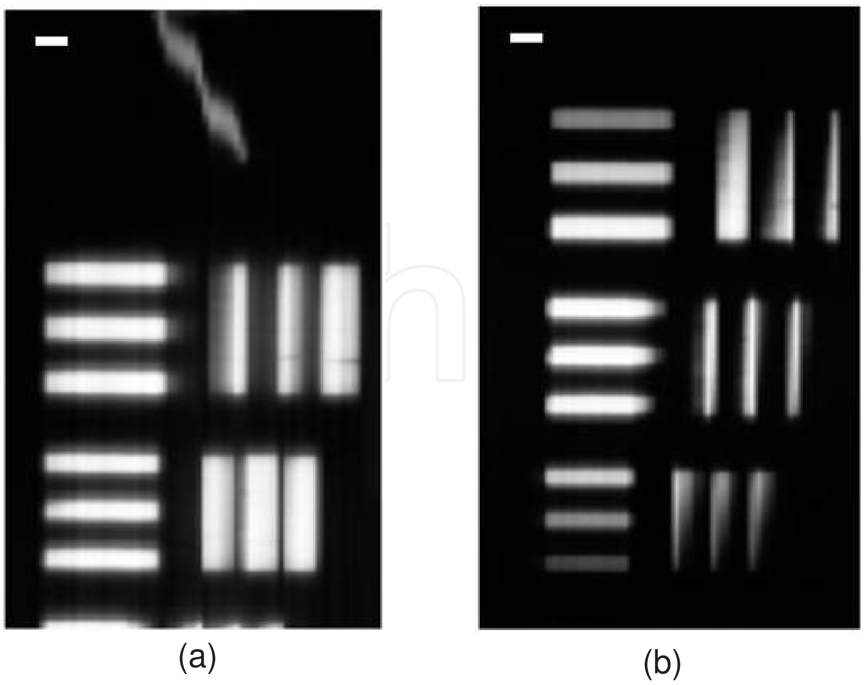


Figure 11. Images of USAF target showing some elements (sets of three lines) of group 7 (higher density). (a) wide-field illumination; (b) line-illumination. Scale bar is 5 µm (horizontal direction – y-axis; vertical direction – x-axis).

This is also the reason why a detailed explanation of how the measured CTF (response to a square input) is converted to MTF (response to a sinusoidal input) is not presented in this chapter. The mathematical expression used for this conversion is:

$$MTF(\xi_f) = \frac{\pi}{4}\sum_{n=0}^{N} B_{2n+1}\frac{CTF\left[2n+1\left(\xi_f\right)\right]}{2n+1} \quad para\ \frac{\xi_f}{2N+3} < \xi \le \frac{\xi_f}{2N+1} \tag{4}$$

where B_{2n+1} is equal to-*1*, *0* or *1*.

Thus the selection of the line is made in an interactive mode using *USAF_image* GUI. When the ROI is exactly as desired the user gives the order to calculate the parameters namely I_{max} and I_{min}, already described and previously shown in equation 1. The range of spatial frequencies used for the determination of MTF is much lower than the range of frequencies the system is able to pass. However this is considered adequate in several applications [4,5,6].

Results are presented in figure 12. In wide-field and line-illumination experimental data range from 32 lp/mm to 228 lp/mm and 128 lp/mm to 228 lp/mm, respectively. Experimental cutoff frequencies (maximum spatial frequency the system is able to pass) are 719 *lp/mm* and 975 lp/mm for wide-field and line-illumination, respectively. It had been used a 40X 0.65 NA objective lens which diffraction-limited (ideal system with no aberrations) cutoff frequency is *1136 lp/mm*.

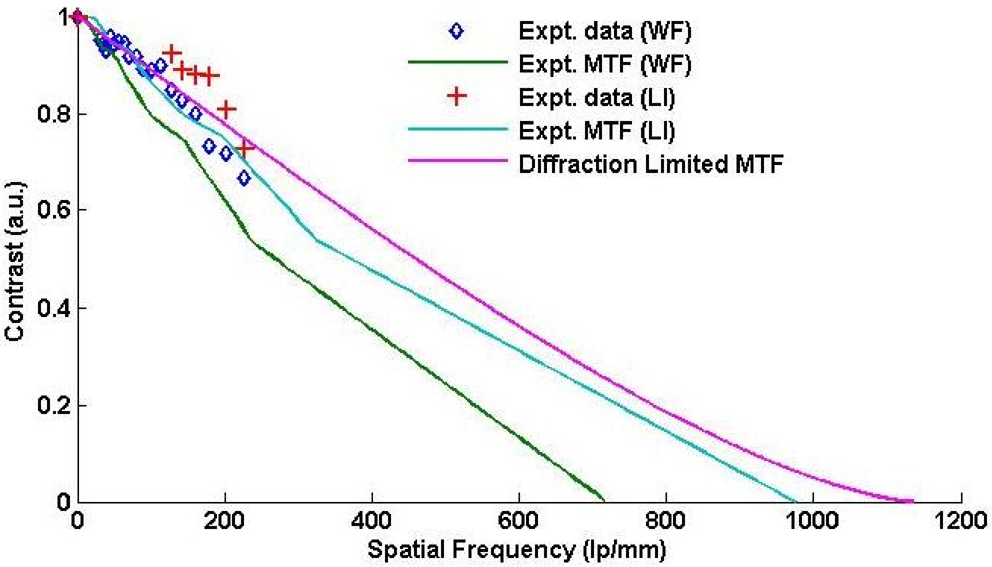


Figure 12. Comparison of wide-field (WF) to line-illumination (LI) for experimental data and calculated diffraction-limited MTF.

5.2.2. Lateral resolution measurement (new method for the determination of PSF)

Images of this bench-microscope have contributions of light coming from planes in front and behind the focal plane. A mathematical model describing the image formation process is used frequently in these cases to remove this blur. From a compromise between data processing demands and required accuracy it had been used one simplified model as follows:

$$\mathrm{I} = H\,O + b \tag{5}$$

where *I* is the image collected in sensor, *H* is the sampled PSF (matrix representing blur) sometimes space variant, *O* is a discrete object and *b* is background light.

To use this model it is necessary to calculate the PSF. In this method that is performed using as object one line of USAF target. It obeys the fundamental condition that object spatial parameters are well-known. So it was applied using lines of higher spatial frequencies in the range from 64 to 228 lp/mm, as those shown previously in figure 11.

For the implementation of this method some code was developed in MATLAB and included in *USAF_image*. It consists on the steps presented in table 3. Owing to anisotropy imposed by the use of a linear image sensor, the PSF in x-axis and y-axis, respectively, parallel and perpendicular to sensor, are different. Thus the method was applied separately for lines in both directions.

This algorithm was applied to obtain PSF_{xx} using a line along x-axis in the group 6 of USAF target. This one-dimensional function was used as image (*I*). The object (*O*), built in MATLAB, was a step function. Its width is 8 μm according to target specifications.

Step #	Description	Function
1.	Selection of one line from the USAF target	$I\,(Image)$
2.	Subtraction of background	$I-b\,,\;b\,(background)$
3.	Calculation of its FFT (*Fast Fourier Transform*)	$fft(I-b)$
4.	Definition of correspondent theoretical line as object	$O\,(Object)$
5.	Calculation of FFT of the object	$fft\,(O)$ fft(O)
6.	Determination of FFT of the PSF (application of model)	$fft(H)=fft(I-b)\,/\,fft(O)$
7.	Calculation of its inverse FFT (corresponds to PSF)	$H=ifft(\,fft(H))$

Table 3. Description of the algorithm executed for the determination of the PSF.

The PSF$_{yy}$ was calculated in a similar mode. In this case an image of a line along y-axis in the group 7 was used with a step function of 3.1 μm width.

For the 40X 0.65NA objective the Nyquist limit is 258 nm. Therefore the minimum sampling interval of 1 μm used in y-axis, which corresponds to the used unit space in the scanning of object stage, is above Nyquist limit. This impairs the result concerning PSF$_{xx}$.

Apart from sampling issues results agree with theory in the sense that PSF_{yy} is narrower than PSF_{xx}. Ratio of lateral resolution in both axes is near the 1.4 factor which is the typical improvement of confocal microscopy. The full width of half maximum of PSF_{yy} is close to 300 nm.

To illustrate and close this subject, in figure 13 it is represented the two-dimensional point spread function in the XY plane. It had been built putting together the one-dimensional PSF curves in each axis.

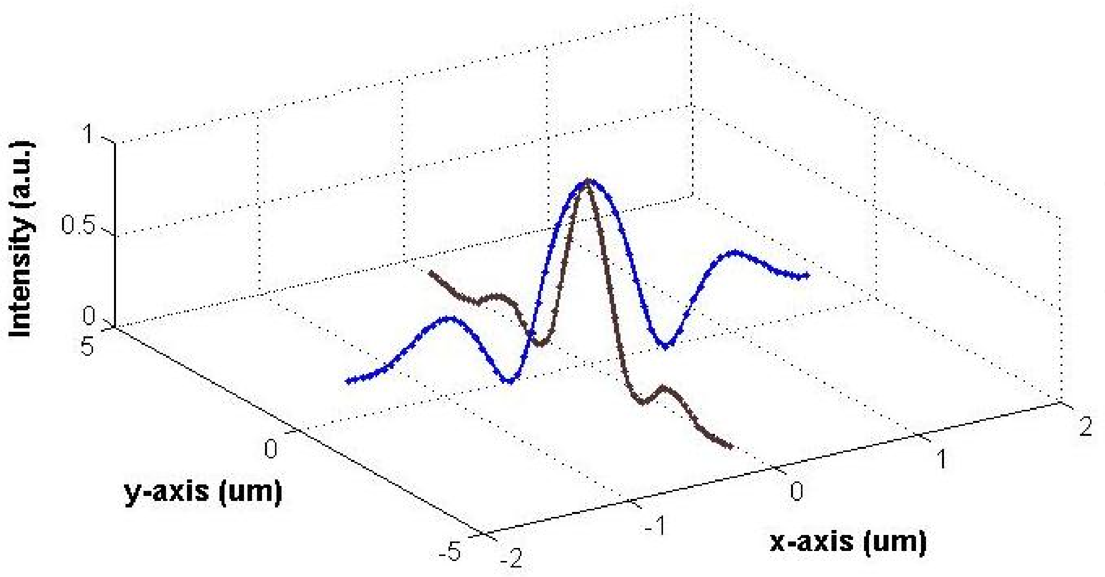


Figure 13. Representation of two-dimensional point spread function (PSF) built from one-dimensional PSF in both x- and y-axis (different scales in the two axes)

5.3. Integrated circuit and printed circuit board inspection

One inspection task in the manufacturing of integrated circuits occurs after the bonding process in which a metallic wire (copper, gold or aluminum) is used to connect a silicon die to package terminals. The diameter of this bonding wire depends on the required specifications of each integrated circuit (IC). Typically it is of the order of magnitude of a few tens of microns.

Another inspection procedure is relative to quality assurance of a printed circuit board (PCB) in order to comply with customer specifications. It consists on the measurement of track dimensions namely its width and thickness.

These inspection processes were performed with this bench-microscope. The graphical user interface of the developed MATLAB application, *IC_image*, is presented in figure 14.

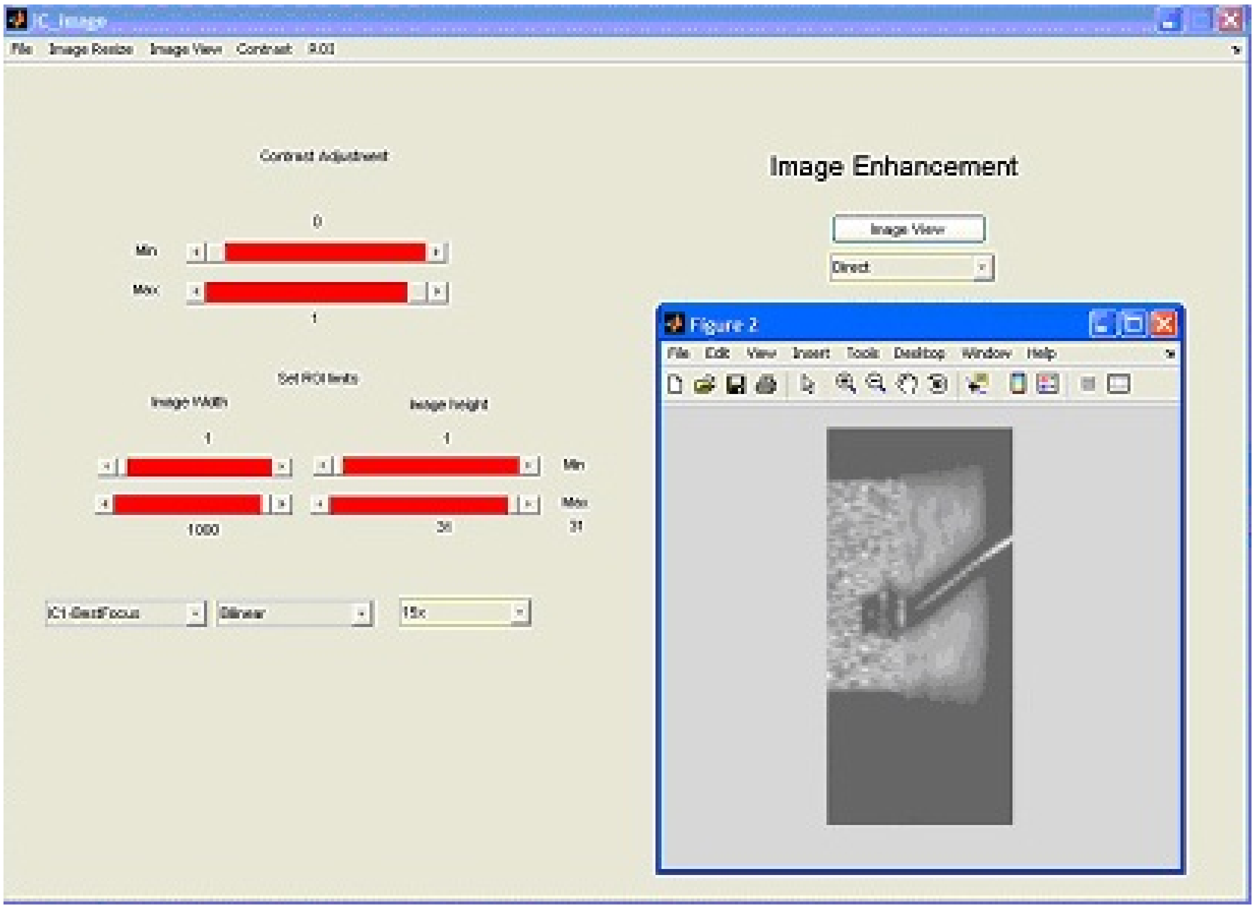


Figure 14. Graphical User Interface – IC_image-developed for the inspection of bonding wire in integrated circuits.

The functionalities of *IC_image* application are summarized below:

- Image settings
  - User definition of ROI limits in the two axes (through the GUI);
  - User definition of contrast adjustment (through the GUI)

- Image visualization
  - User selection of the set of sensor images previously acquired (through the GUI);
  - User definition of the *imresize* method and parameter (through the GUI);
  - User selection of the visualization mode (through the GUI);
  - Visualization of 2D or 3D images

5.3.1. Determination of bonding wire diameter

As it will be shown later, for the determination of wire diameter as well as PCB track dimensions it was necessary to implement algorithms for the reconstruction of 3D images. Then besides the operations for image visualization it had been necessary to include these algorithms in the *IC_image* application.

Two of these algorithms will be presented below. Algorithm #1 is more simplified and consists on the determination of the maximum of intensity for each pixel through a stack of 2D images. Then an image, called *auto-focus*, is built using these pixel values which means that presumably the most focused XY plane is find for each pixel. Therefore these images contain 3D information through the application of:

$$I(x_i, y_i) = \underset{k=1}{\overset{N}{MAX}}\; I(x_i, y_i, z_k) \tag{6}$$

Algorithm #2 adapted from literature [7] had been also used to remove light from planes in front and behind the focal plane. It is based on the assumption that ideally at the focal plane the light is focused on one point and as it departs from that plane it spreads over an increasing area. The mathematical expressions are:

$$\begin{gathered} I_{ext}(x_i, y_i) = \underset{k=1}{\overset{S}{MAX}} \left[\frac{\sum\limits_{a=-(N-1)/2}^{(N-1)/2} \sum\limits_{b=-(M-1)/2}^{(M-1)/2} \left(I_m(x_i, y_i, z_k) - I(x_i + a, y_i + b, z_k) \right)^2}{I_m{}^2(x_i, y_i, z_k)} \right] \\ I_m(x_i, y_i, z_k) = \frac{1}{(N \cdot M)} \sum\limits_{a=-(N-1)/2}^{(N-1)/2} \sum\limits_{b=-(M-1)/2}^{(M-1)/2} I(x_i + a, y_i + b, z_k) \end{gathered} \tag{7}$$

where:

- *N* and *M* are spatially parameters used for the definition of a ROI to be considered around each pixel. These values had been established from specific geometrical and sampling parameters of this bench-microscope. *N* and *M* represent the number of pixels and linear images, respectively.

- *Inda* and *indb* are adjustable and obey the requirement of $-0.5 \le inda,\ indb \le 0$. After some tests, in this inspection task middle values had been used ($inda = indb = -0.25$). S is the amount of 2D images to be used.

The output of this algorithm #2 provide data that may be presented through three visualization modes. So the user may select in *IC_image* GUI one of the following modes:

1. Processed image $I_{ext}(x_i,\ y_i)$ (matrix values obtained directly with the mathematical expression in equation 7)
2. Raw-data image $I(x_i,\ y_i,\ z_k)$ where z_k is the position along z-axis in which the pixel values are maxima in each pixel of the processed image, $I_{ext}(x_i,\ y_i)$.
3. Processed topographic map $h(x_i,\ y_i)$ representing the z_k values already described in the previous mode.

Image in visualization mode (2) is also an *auto-focus* image as the image obtained by equation 6. These two images are shown in figure 15 for one example where six two-dimensional images (parameter S in equation 6 and equation 7) and a ROI (defined by N x M in equation 7) equal to 17 x 1 had been used.

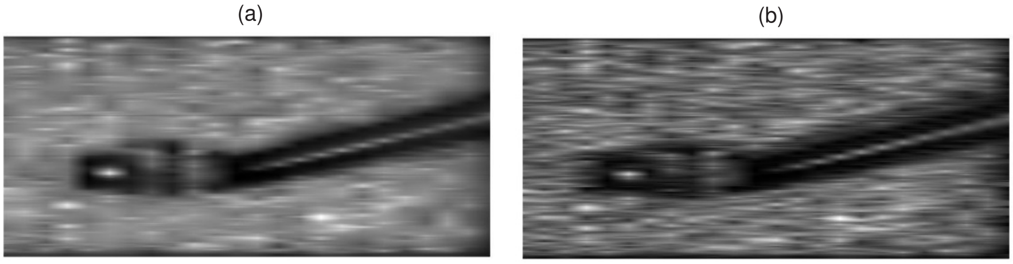


Figure 15. Auto-focus images: (a) algorithm #1 (from Equation 6); (b) algorithm #2 (from Equation 7) using visualization mode (2). FOV: 460 µm x 210 µm (horizontal – y-axis; vertical – x-axis)

In a very similar mode in figure 16 are presented the respective 3D maps. It means that the data shown in part (a) had been obtained from algorithm #1 and the visualization is similar to mode (3).

For the determination of the gold wire diameter the following three images had been used, for comparison purposes:

1. Raw-data auto-focus image $I(x_i,\ y_i,\ z_k)$ where z_k is the position along z-axis in which the pixel values are maxima in each pixel of the raw-data image $I(x_i,\ y_i)$.
2. Processed auto-focus image $I(x_i,\ y_i,\ z_k)$ where z_k is the position along z-axis in which the pixel values are maxima in each pixel of the processed image $I_{ext}(x_i,\ y_i)$.
3. Processed height image such as $I(x_i,\ y_i,\ z_k) = h(x_i,\ y_i)$ where $h(x_i,\ y_i)$ represents the z_k values already described for the previous image.

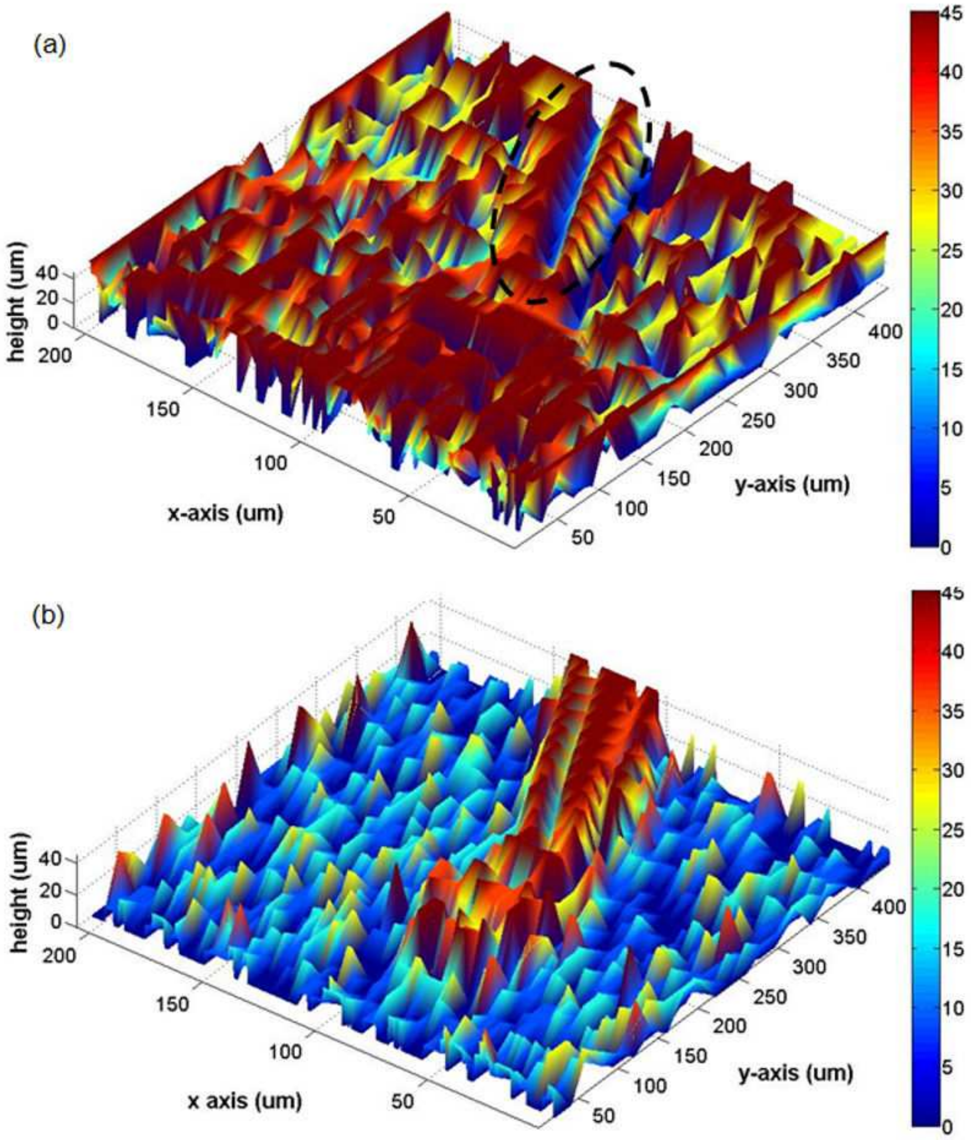


Figure 16. 3D maps achieved using visualization mode (3): (a) algorithm #1; (b) algorithm #2. Dashed oval in part (a) surrounds a linear structure with the same orientation of the structure that is extremely clear in part (b).

In this image (3) shown in figure 17. (a) as well as in images (1) and (2) both shown in figure 15 a ROI with 300 pixels (117 μm) and twelve linear images (220 μm) had been defined. This ROI should contain a part of the wire which is away from the package terminal. In this manner the diameter measurement is not affected by the stress imposed in the wire to make the connection.

However this implies that the MATLAB application must use a 3D reconstruction algorithm to obtain a focused image of this part of the wire because its height is not constant. This ROI in the three images (1) to (3) is shown in figure 17 (b) to (d).

The aim is to develop a method that for each linear image in the ROI is able to find the amount of pixels that belong to wire. So this algorithm is adjustable depending on the image. For images

(1) and (2) it calculates the number of pixels with intensity lower, and in image (3) higher, than a pre-defined threshold.

Using a threshold value of 50% it corresponds to the calculation of FWHm (Full Width at Half minimum) and FWHM, respectively. A geometric correction factor should be used as the wire is not horizontal in image because it is not perpendicular to sensor.

The results of the application of this algorithm on the three images are presented in table 4. Besides average values of wire diameter its standard deviation was also calculated. The diameter from image (2) is considerably larger than from the other two images. The definition of a threshold different than 50% would mitigate this difference induced by lower image contrast in this particular image.

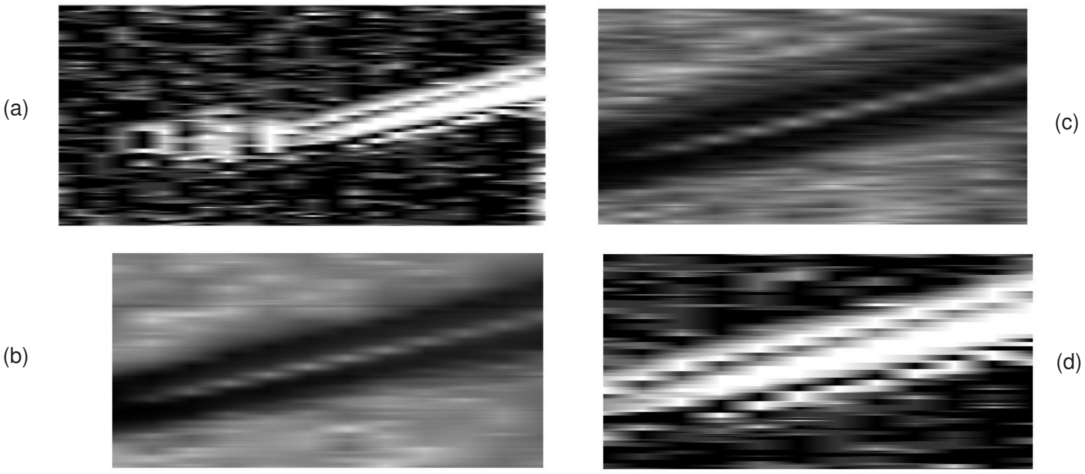


Figure 17. (a) Complete image (3). FOV: 460 µm x 210 µm. (b) to (d) images of the ROI extracted from images (1) to (3), respectively. FOV: 220 µm x 117 µm (horizontal – y-axis; vertical – x-axis). Image in part (a) uses a different scale.

Image	Wire diameter (µm)	
	Average	Std. deviation
(1)	43,06	4,51
(2)	61,78	6,83
(3)	36,24	7,97

Table 4. Wire diameter average and standard deviation values calculated from images (1) to (3) using the ROIs shown in figure 17. (a) to (c), respectively.

5.3.2. Determination of width and thickness of PCB tracks

Both track orientations, parallel and perpendicular to sensor, had been used. However it was accomplished using a very similar method. For illustration it is presented the parallel one in which twelve two-dimensional images had been acquired.

Using the equation 6 it was calculated the *auto-focus* image, designated previously by image (1). This image is shown in figure 18. Also the topographic (3D) map $h\,(x_i,\ y_i)$ representing the z_k values that correspond to the position along z-axis in which the pixel values are maxima in each pixel of the raw-data image $I(x_i,\ y_i)$. To represent a two-dimensional (2D) profile it was selected the pixel n of the sensor and $h\,(x_n,\ y_i)$ is the height of that pixel n for each sensor line.

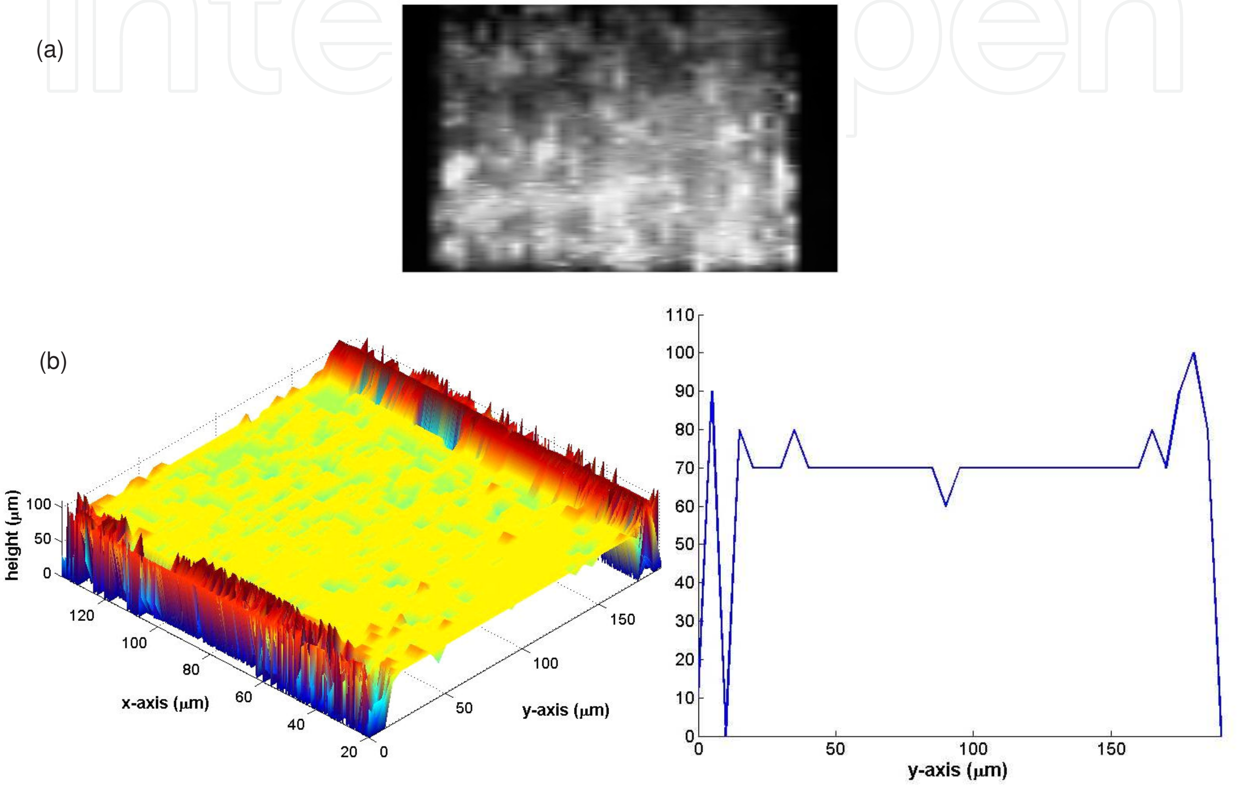


Figure 18. PCB track parallel to the sensor. (a) raw-data *auto-focus* image. FOV: 190 μm x 115 μm (horizontal – y-axis; vertical – x-axis). Same scale in both axes. (b) raw-data topographic (3D) map and a 2D profile drawn for pixel *n*.

In spite of the ringing effects present at track borders, that hamper the measurement of its width and reduce precision and accuracy, an algorithm had been developed for its calculation. This algorithm implemented in *IC_image* application consists on the following steps:

1. Definition of a ROI completely inside the track;

2. Determination of height $h\,(x_i,\ y_i)=z_k$ where z_k is the position along z-axis in which the pixel values are maxima for each pixel of the raw-data image $I(x_i,\ y_i)$.

3. Determination of height mean (h_{mean}) and standard deviation (h_{std}).

4. Definition of the lower (h_{min}) and upper (h_{max}) height limits inside the track. The considered limits had been: $h_{\substack{\max \\ \min}} = h_{mean} \pm h_{std}$.

5. Definition of other ROI including the complete field of view along y-axis (perpendicular to the track);

6. Calculation of the number of lines (position in y-axis) in which the height lays in the range $h_{\min} \leq h \leq h_{\max}$ (over all the pixels inside the ROI);

7. Determination of track width mean and standard deviation expressed in number of lines;

8. Conversion to microns and mils.

The height value $h(x_i, y_i)$ represents the axial position of the track. Therefore the track height is easily calculated through the determination of the height of the exterior side plane and its subsequent subtraction. This calculation had been made for a track perpendicular to sensor but this example is not covered in this chapter. Table 5 shows the results of the implementation of this algorithm.

Track parameter	Mean	Std. deviation
Height	4,2890*	0,5963*
Width	120,40 µm (4,74 mils)	9,74 µm (0,38 mils)

Table 5. Mean and standard deviation values of the track height and width obtained using a ROI inside the track and over the complete y-axis, respectively. (*) rigorously it indicates only the axial plane and not the height.

5.3.3. Application in profilometry

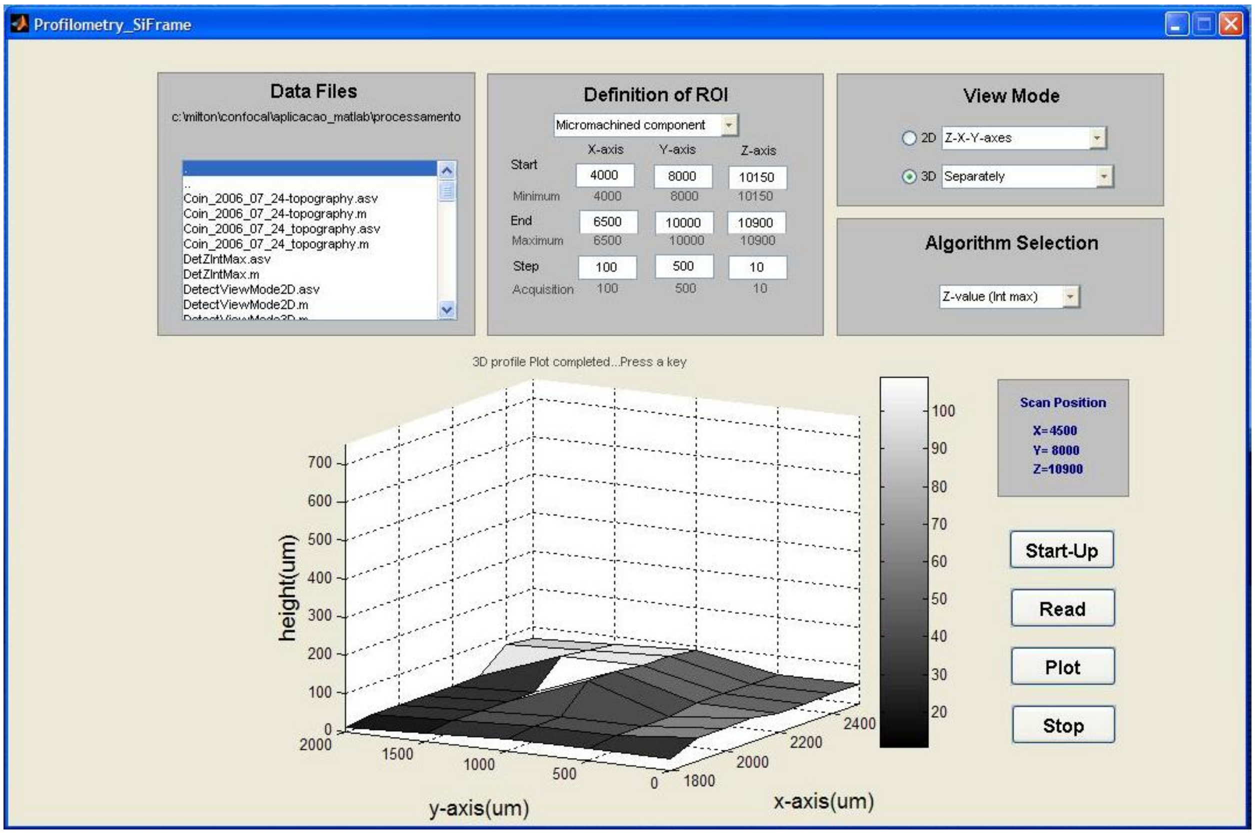
Owing to its depth discrimination ability, optical techniques as confocal microscopy have been used in profilometry. It consists on build three-dimensional profiles in order to measure different surface or geometrical dimensions such as roughness, height / depth or width.

Another MATLAB application, *Profilometry_SiFrame*, which GUI is presented in figure 19 had been developed.

This MATLAB application has the following functionalities:

- Profile settings
  - User definition of ROI limits in the two axes (through the GUI);
- Profile visualization
  - User selection of the set of sensor images previously acquired (through the GUI);
  - User selection of the visualization mode (2D or 3D) (through the GUI);
  - User definition of the scanning sequence to be used in 2D visualization (through the GUI);
  - User selection of the 3D reconstruction algorithm (through the GUI);
  - Interactive visualization of a sequence of 2D profiles (stop by an user order).

- Visualization of 3D profiles



Profilometry_SiFrame

Data Files

c:\milton\confocal\aplicacao_matlab\processamento

.
..
Coin_2006_07_24-topography.asv
Coin_2006_07_24-topography.m
Coin_2006_07_24_topography.asv
Coin_2006_07_24_topography.m
DetZIntMax.asv
DetZIntMax.m
DetectViewMode2D.asv
DetectViewMode2D.m
DetectViewMode3D.m

Definition of ROI

Micromachined component

	X-axis	Y-axis	Z-axis
Start	4000	8000	10150
Minimum	4000	8000	10150
End	6500	10000	10900
Maximum	6500	10000	10900
Step	100	500	10
Acquisition	100	500	10

View Mode

○ 2D Z-X-Y-axes
● 3D Separately

Algorithm Selection

Z-value (Int max)

3D profile Plot completed...Press a key

Scan Position
X=4500
Y= 8000
Z=10900

Figure 19. Graphical User Interface – *Profilometry_SiFrame* – developed for building and visualization of micromachined component profiles.

A micromachined component that contains a three-dimensional silicon frame had been the test object. For the assessment of the quality of the results, a scheme with frame specification is also presented in figure 20.

Experimental profiles shown in figure 20 are a raw-data topographic (3D) map and a 2D profile where height is represented versus x-axis (sensor orientation). This profile had been drawn from topographic map in (b) representing one of the lines across the frame. Experimental values of width, height or slope of silicon frame are superimposed in the 2D profile.

The lateral side of the silicon frame had been aligned with y-axis. Thus lateral walls of the frame are perpendicular to sensor. On the other hand three different ROI were defined over x-axis. In ROI separation the slope of the frame wall is higher than its maximum value considering the particular objective used. Owing to this slope the light reflected in frame surface is not gathered by the objective and consequently it is not collected in the sensor.

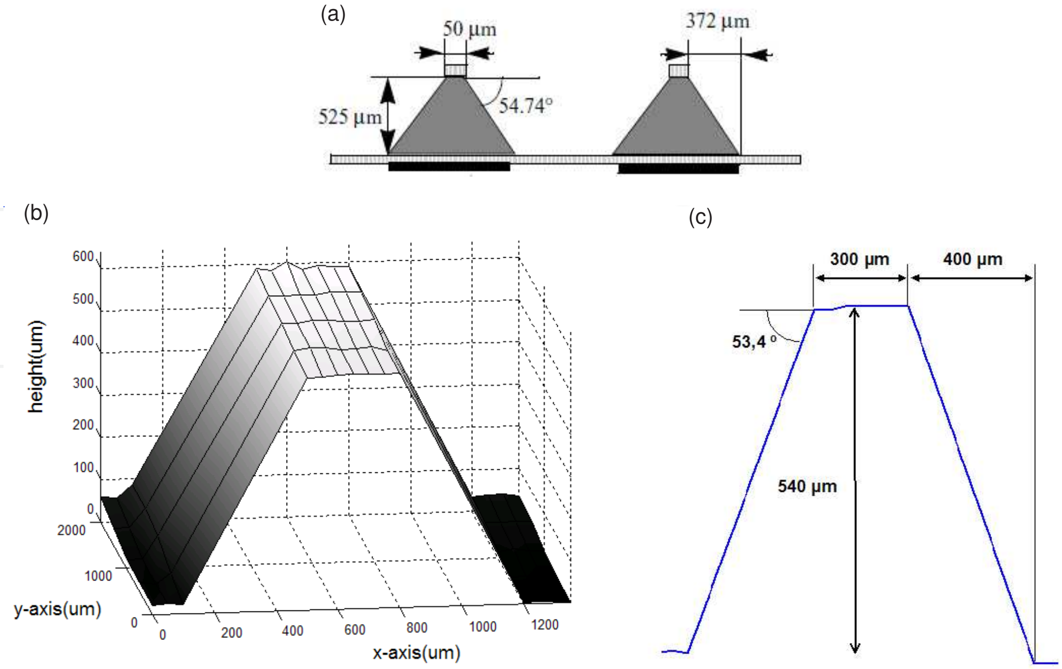


Figure 20. (a) Scheme of the silicon frame illustrating its dimensions (reproduced from [8]). (b) experimental 3D profile of the frame (x-and y-axis represent parallel and perpendicular directions relative to sensor, respectively). (c) 2D profile showing the results for frame dimensions (not to scale).

6. Conclusion

The challenge of building a laboratorial prototype of a microscope using a linear image sensor had been successfully attained. It had been shown its ability to perform the acquisition and visualization of images containing three-dimensional information as well as its potential application in materials science field.

MATLAB has played a fundamental role for this outcome. Apart from the optical layout every task since sensor readout through communication of sensor data to the computer for image visualization to image reconstruction algorithms had been implemented in MATLAB applications. In this manner the result is a bench-microscope completely controlled by a computer user. Figure 21 summarizes overall functions implemented in the four MATLAB applications in order to achieve this computer-controlled platform.

The essential graphical user interface had been developed in a much faster and easier mode using functions of MATLAB core than previously with Borland C. Besides it is important to emphasize the versatility provided by its toolboxes. Particularly *Instrument Control toolbox* that made the implementation of all acquisition and control tasks, commanded from the computer through its serial port, look like a trivial task. Also *Image Processing toolbox* had shown how the implementation of reconstruction algorithms and of any image related operation may be performed in an easy way despite the complexity that people might assume because of matrix approach. In summary, with every sense, it is a MATLAB-based microscope.

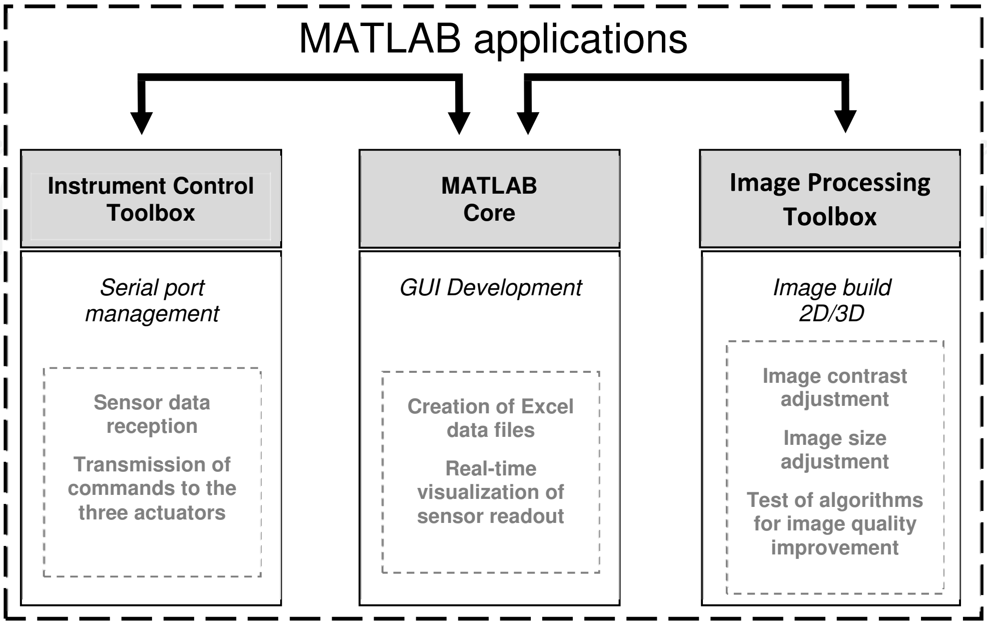

MATLAB applications
Instrument Control Toolbox
MATLAB Core
Image Processing Toolbox
Serial port management
Sensor data reception
Transmission of commands to the three actuators
GUI Development
Creation of Excel data files
Real-time visualization of sensor readout
Image build 2D/3D
Image contrast adjustment
Image size adjustment
Test of algorithms for image quality improvement

Figure 21. Diagram of the functionalities of the MATLAB applications.

Author details

Milton P. Macedo[1,2]

1 Instituto Politécnico de Coimbra, ISEC, DFM, Coimbra, Portugal

2 Centro de Instrumentação, Department of Physics, University of Coimbra, Coimbra, Portugal

References

[1] Murphy DB. Fundamentals of light microscopy and electronic imaging. USA: Wiley-Liss; 2001.

[2] Lee J, Rogers JD, Descour MR. Imaging quality assessment of multi-modal miniature microscope. Optics Express 2003; 11(12) 1436-1451.

[3] Boreman GD, Yang S. Modulation transfer function measurement using three-and four bar target, Applied Optics supplement 1995; 8050-8052.

[4] Williams TL. The Optical Transfer Function of Imaging Systems. USA: IoP; 1999.

[5] Williams CS, Becklund OA. Introduction to the Optical Transfer Function. USA: SPIE Press; 2002

[6] Sabharwal S et al. Slit-scanning confocal microendoscope for high-resolution in vivo imaging. Applied Optics 1999; 38 (34) 7133-7144.

[7] Tympel V. Three dimensional animation with a conventional light microscopy. In: Cogswell CJ, Conchello JA, Wilson T. (eds) BiOS1997: proceedings of three-dimensional microscopy: image acquisition and processing IV, 12-13 February 1997, San Jose, USA. Washington: SPIE Press; 1997. p190-198.

[8] Correia JH, Bartek M, Wolffenbuttel RF. Load-deflection of a low-stress SiN-membrane/Si-frame composite diaphragm. In: First International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators: conference proceedings, 6-8 April 1998, Santa Clara Marriott, CA, USA. Boston: Computational Publications; 1998. p563-568.