PiLoupe

Simple Measuring Microscope for a Watchmaker's Lathe

 Adrian Rawson © 2021

Introduction

This project was conceived through the need to make a small component for an old watch realising that much of the work must be performed under magnification.

The decision was taken to buy a cheap digital USB connected microscope manufactured in China as a first step.

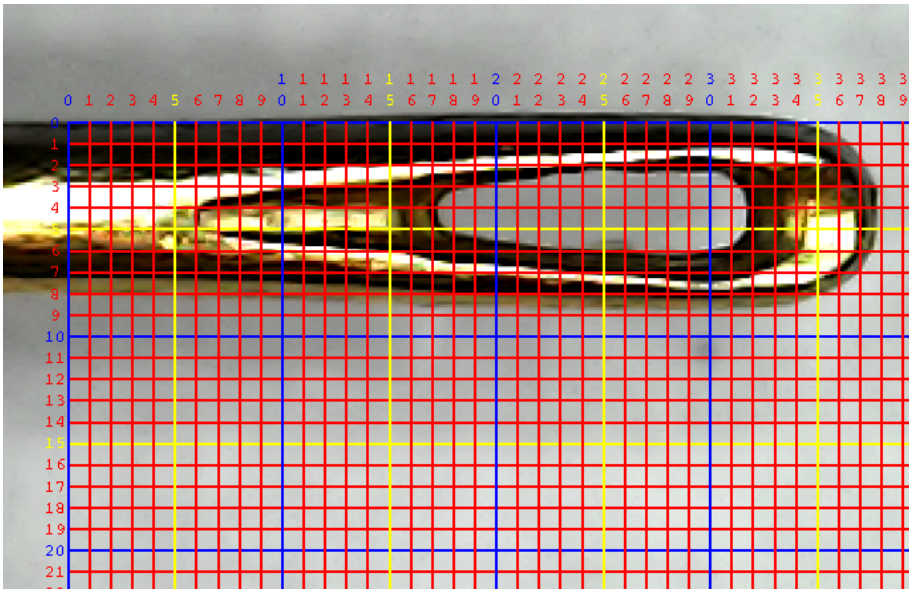
This claimed to provide a magnification of 50X to 1600X, but I'm not sure on what this claimed was based. However the device has proven to be very flexible in its use in that it can be positioned at any distance from an object and focused. Obviously the closer one gets the greater the magnification achievable. The one I bought was also fitted with adjustable LED illumination, a snapshot button and downloadable software.
The software included a facility to take measurements during a short video recording session, but this didn't seem to offer constant measuring during a live video stream which I considered necessary for my needs.

 Adrian Rawson © 2021

What I did next was write some software to run on a Raspberry Pi that provided the facilities I wanted to help make the watch component.

Please note that all the screen images shown in this document were created by running PiLoupe on a Mac computer.

After some trial and error I succeeded in superimposing a scale on the video stream, and at the magnification achievable, and a workable distance, the scale gave a resolution of 100µm (i.e. tenth of a mm). See the picture below of a sewing needle which has a diameter of 0.8mm.



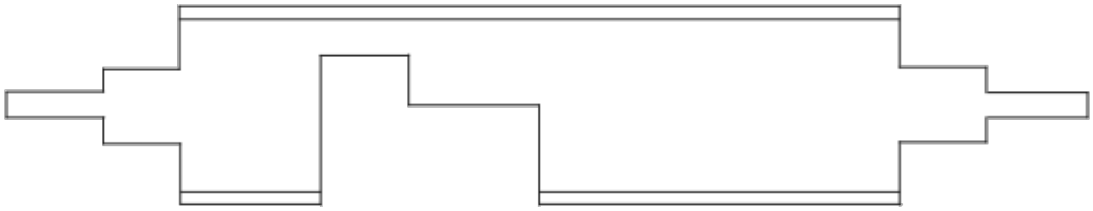
The scale is set according to a pre-measured value. In this case the diameter of the needle is known, so the scale is zoomed until it shows a dimension of 0.8mm across the needle. Once the scale is set the eye of the needle can be seen to measure 0.4mm across its width and 1.5mm along its length.

Having successfully superimposed a scale on the video stream, the next challenge was to superimpose a representation of the part I was trying to make.

When turning a copy of a worn out balance-staff it seems common practice to offer up the old part as a guide. However the old part I was trying to make was a cylinder escapement which was broken and partially lost. So I decided to use CAD software to draw the part from measurements taken from the remaining bit and the watch it came from.

 Adrian Rawson © 2021

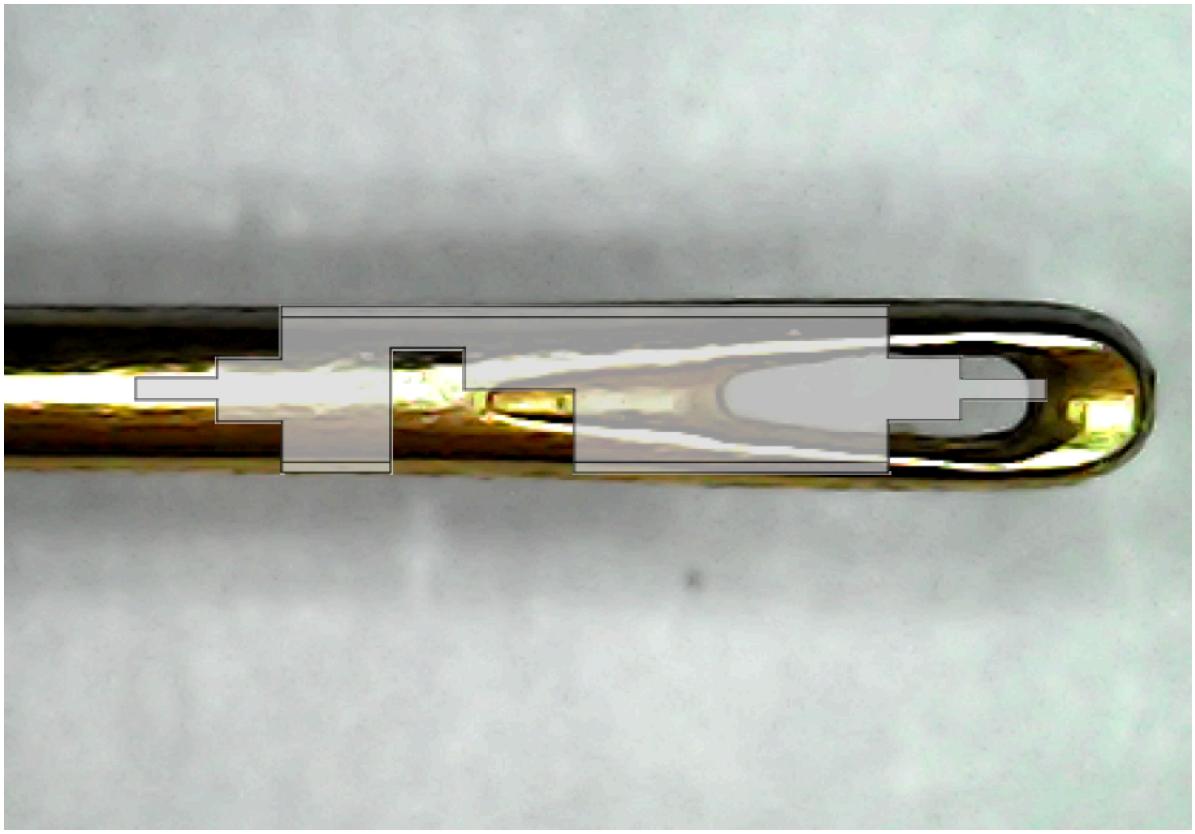
The picture below shows a CAD drawing of the cylinder escapement.



This has a diameter of 0.8mm with a total length of 4.37mm.

The CAD drawing was exported as a ‘png’ file and then with the aid of a graphics editor the background was made transparent, leaving the component white. Also the image was cropped to just include the component detail with as little background as possible.

The software was then enhanced to enable this picture to be zoomed to size and superimposed on the video stream. And subsequently the facility was added to make the image semi-transparent.



 Adrian Rawson © 2021

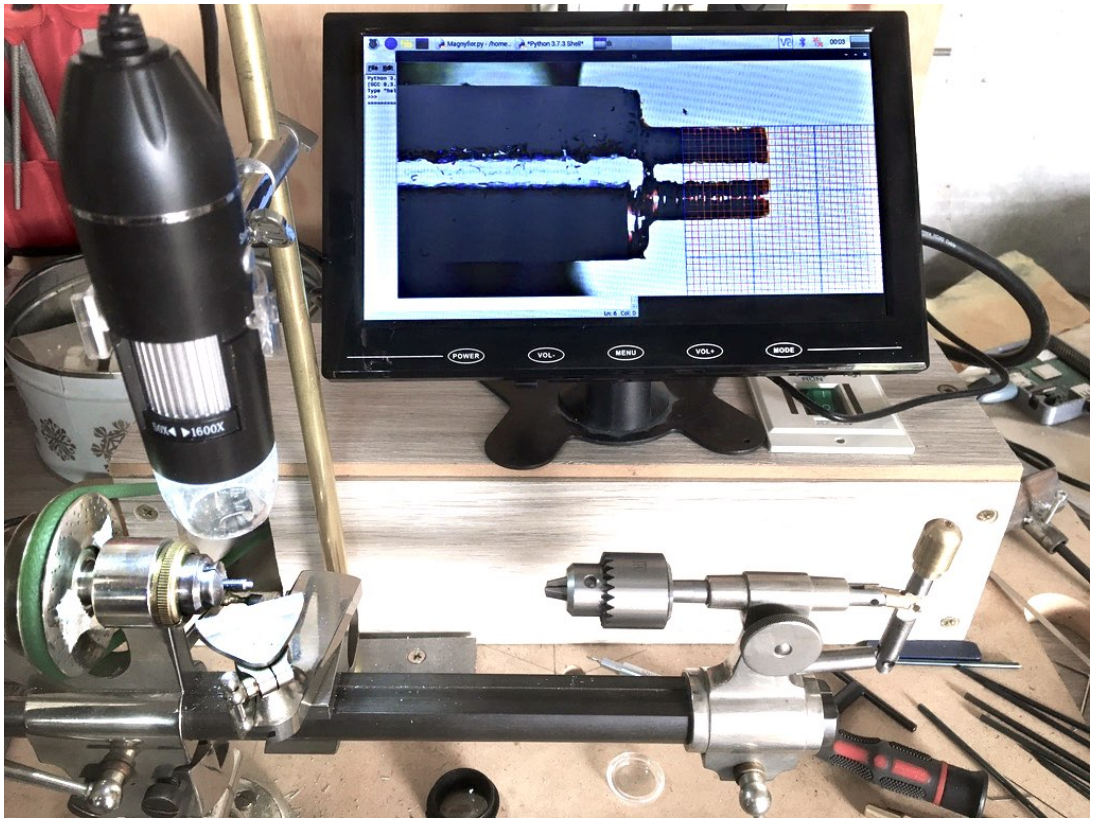
Software/Hardware

As I said in the introduction, this was written to run on a Raspberry Pi. I used an RPi 3A+ which is probably the minimum spec required to give a reasonable performance on the video stream using the USB connected camera. But, more importantly, this is what I had available at the time.

The software is written in Python and uses opencv to manage the video, PIL to manipulate the images and Tkinter to handle the user interface.

It was an important requirement for me not to rely on any keyboard input to control the application, so everything is controlled with a USB connected mouse. In the case of the RPi 3A+ this necessitated the addition of a USB hub to connect both the mouse and the camera. Also needed is an HDMI connected display screen and a power supply. As with many RPi projects the interconnecting wires form a significant part of the solution.

Shown below is the setup in use on my watchmaker's lathe.



 Adrian Rawson © 2021

PiLoupe

User Guide

Running PiLoupe

The application is distributed as a bundled application. When PiLoupe is run it expands its python environment into a directory called _MEIxxxxxx (where xxxxxx is a random number). This can take several seconds.
When PiLoupe closes the support directory is deleted. If however PiLoupe crashes for any reason the support directory may remain, in which case it should be deleted manually.
After initial download PiLoupe will need to be made executable. To do this in File Manager select file properties>permissions and set execute to anyone. Or in Terminal go to the directory where you've downloaded it and type

'**chmod +x PiLoupe**'

On the Raspberry Pi, PiLoupe may be run either from the file manager or from a terminal,

File manager:- Clicking on the PiLoupe application invites you to Execute it or Execute it in a Terminal.

Terminal:- First start a terminal session then issue the command

'./PiLoupe' (from the directory where PiLoupe is stored))

this will run the application using defaults, but you can also pass parameters.

'./PiLoupe -h' gives the result:

```
usage: PiLoupe [-h] [-x <screen_width>] [-y <screen_height>]
               [-c <camera_source>]

optional arguments:
  -h, --help          show this help message and exit
  -x <screen_width>   Width of screen in pixels
  -y <screen_height>  Height of screen in pixels
  -c <camera_source>  Camera source (1 or 2)
```

The default screen width and height are taken from the RPi's screen parameters and these are displayed in Terminal mode. These can be overridden using the '-x' and '-y' options.

 Adrian Rawson © 2021

The camera source default is ‘-1’ which connects to the first camera available. This can be overridden using the ‘-c’ option for a multi-camera system. Camera ‘0’ is also the first camera.

e.g.

./PiLoupe -x640 -y480 -c1

Will set the application screen to 640 pxels wide by 480 pixels high and use the second camera.

N.B. Setting the screen’s height to a value greater than the default height may effect the application’s behaviour.

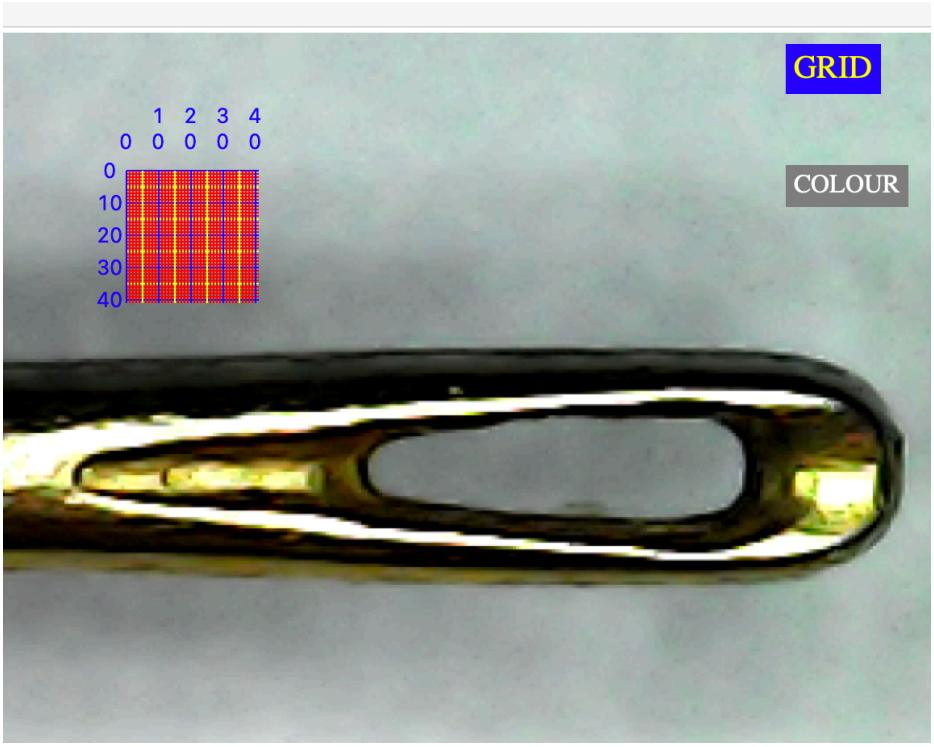
When the program is first started it shows the video stream from the camera and a blue **GRID** button

This indicates that **GRID** mode is selected but initially the measuring grid isn’t displayed. If the **GRID** button is clicked it changes to **IMAGE** which indicates that **IMAGE** mode is selected, but again, initially no image is displayed. The graphic below shows a magnified sewing needle.

GRID Mode

To show the measuring grid right click the mouse to display a pop-up menu and select the first item '**Show/Hide Grid**'.

The grid is displayed and a **COLOUR** button appears.



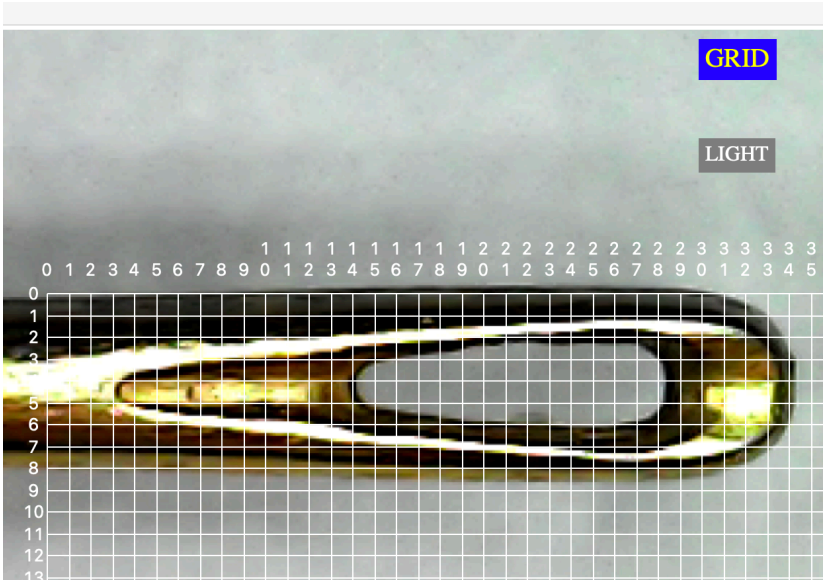
 Adrian Rawson © 2021

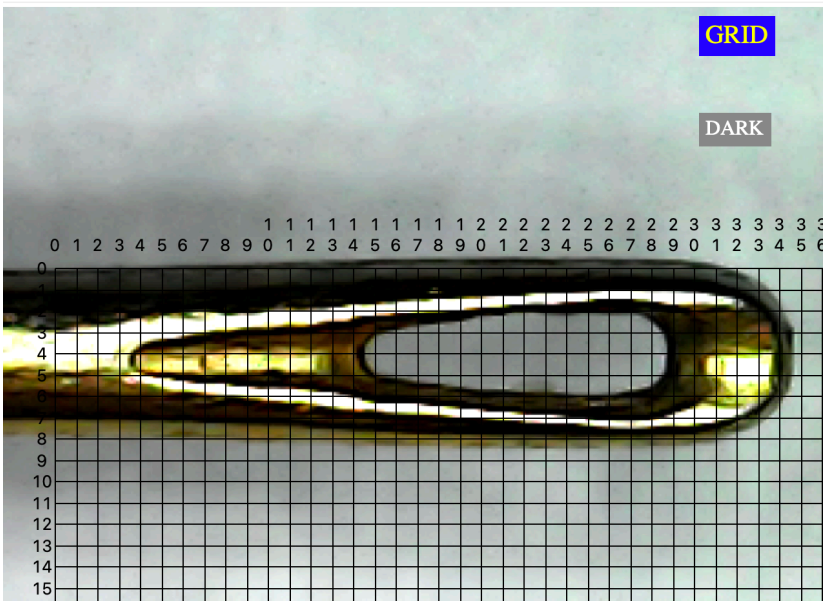
PiLoupe

To position the grid on the screen move the cursor to the target location and click the left mouse button.

To expand the grid click and hold the left mouse button and drag it vertically.

To change the grid colour left click the **COLOUR** button. This will cycle through **COLOUR**, **LIGHT**, **DARK**.

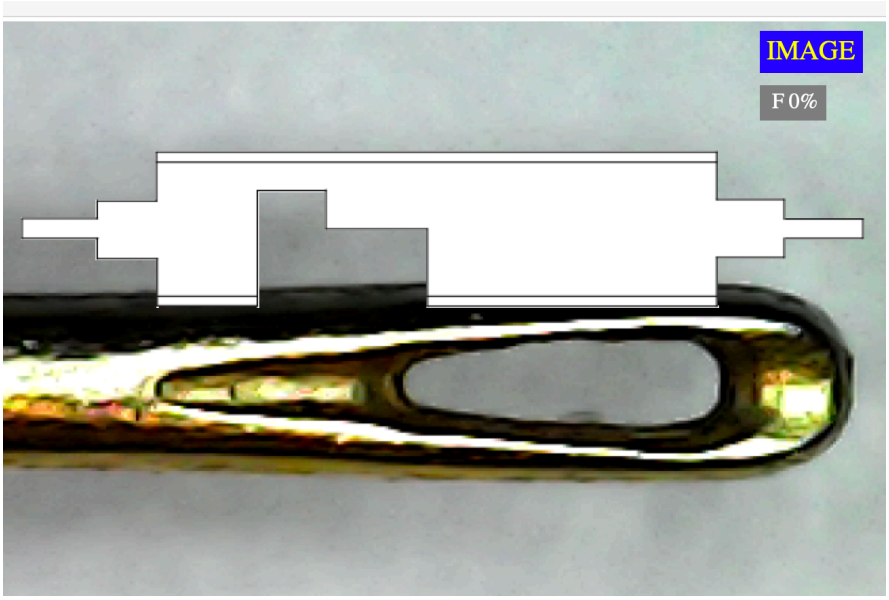




 Adrian Rawson © 2021

IMAGE Mode

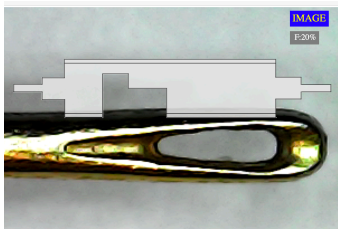
Initially no image is loaded so there is nothing to display. To display an image right click to display the popup menu then select '**Show/Hide Image**'. If an image is already loaded it will be displayed, otherwise a file dialog will open showing files in the subdirectory '/home/pi/images'. Selecting a file will display the image and switch to **IMAGE** mode. Also an '**F 0%**' button will appear.

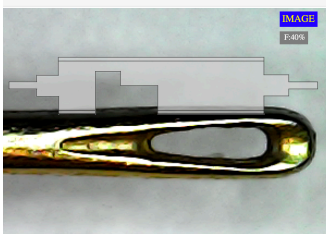


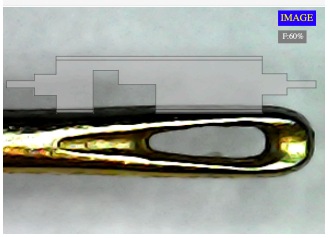
To position the image on the screen move the cursor to the target location and click the left mouse button. The image origin is always at its top left corner.

To expand or shrink the image click and hold the left mouse button and drag it vertically.
N.B. If the image is zoomed in and out repeatedly its definition will suffer.

To change the image transparency left click the **F(ade)** button. This will cycle through **0%**, **20%**, **40%**, **60%**.

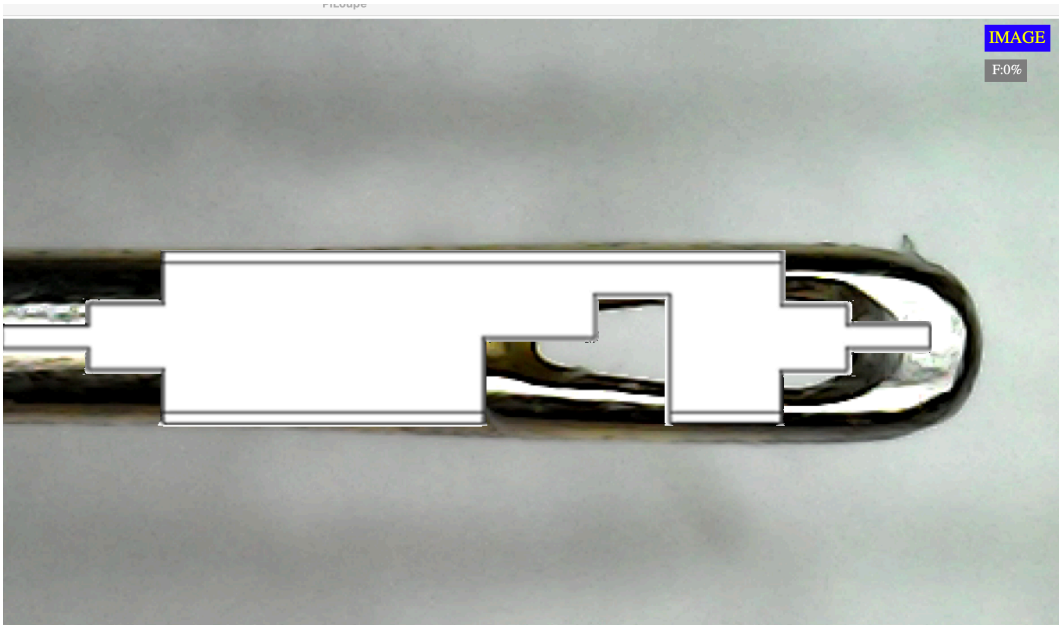




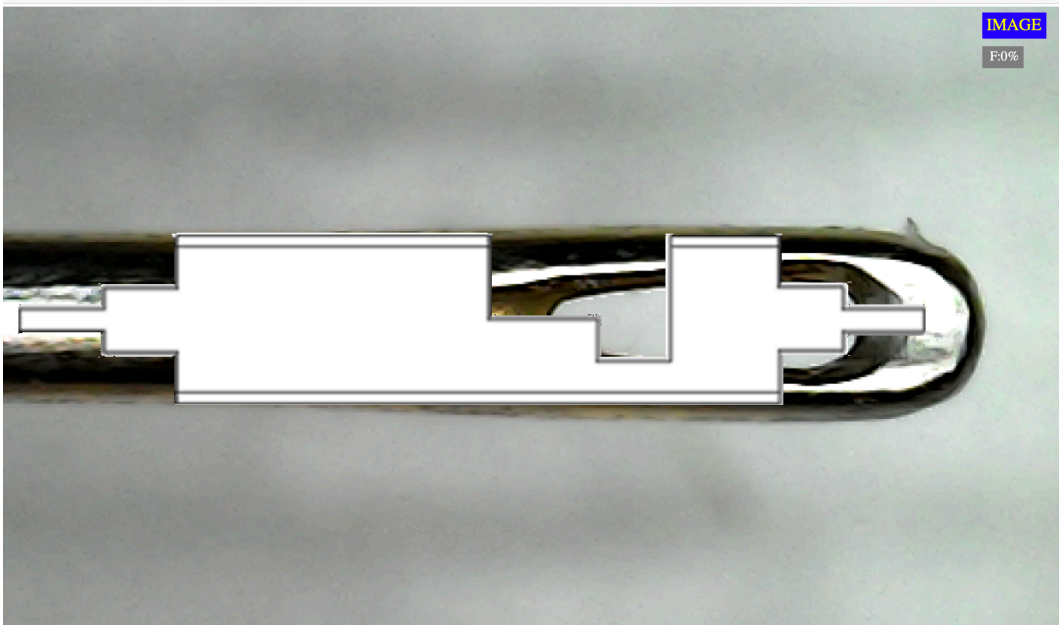


 Adrian Rawson © 2021

To flip the image right/left select '**Flip Image <>**' from the pop up menu.



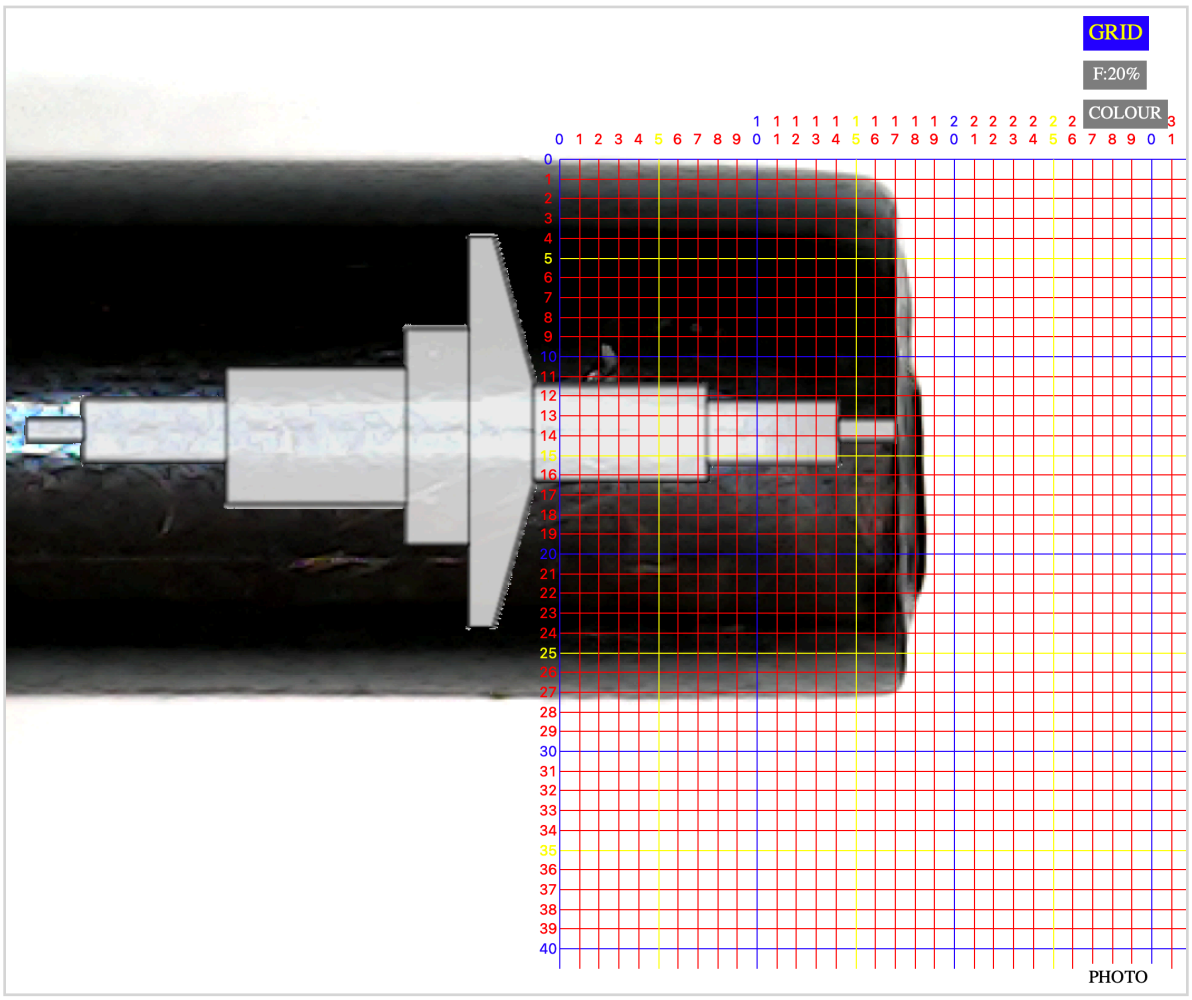
To flip the image top/bottom select '**Flip Image ^v**' from the pop up menu.



 Adrian Rawson © 2021

PiLoupe

If necessary the image and a grid can be displayed at the same time whereby both the **F(ade)** and **COLOUR** buttons will be active. However to move/zoom either the grid or the image the relevant mode must be showing on the blue mode button. Below is a drawing of a balance staff superimposed on 2.73mm diameter blue steel with both image and grid showing.



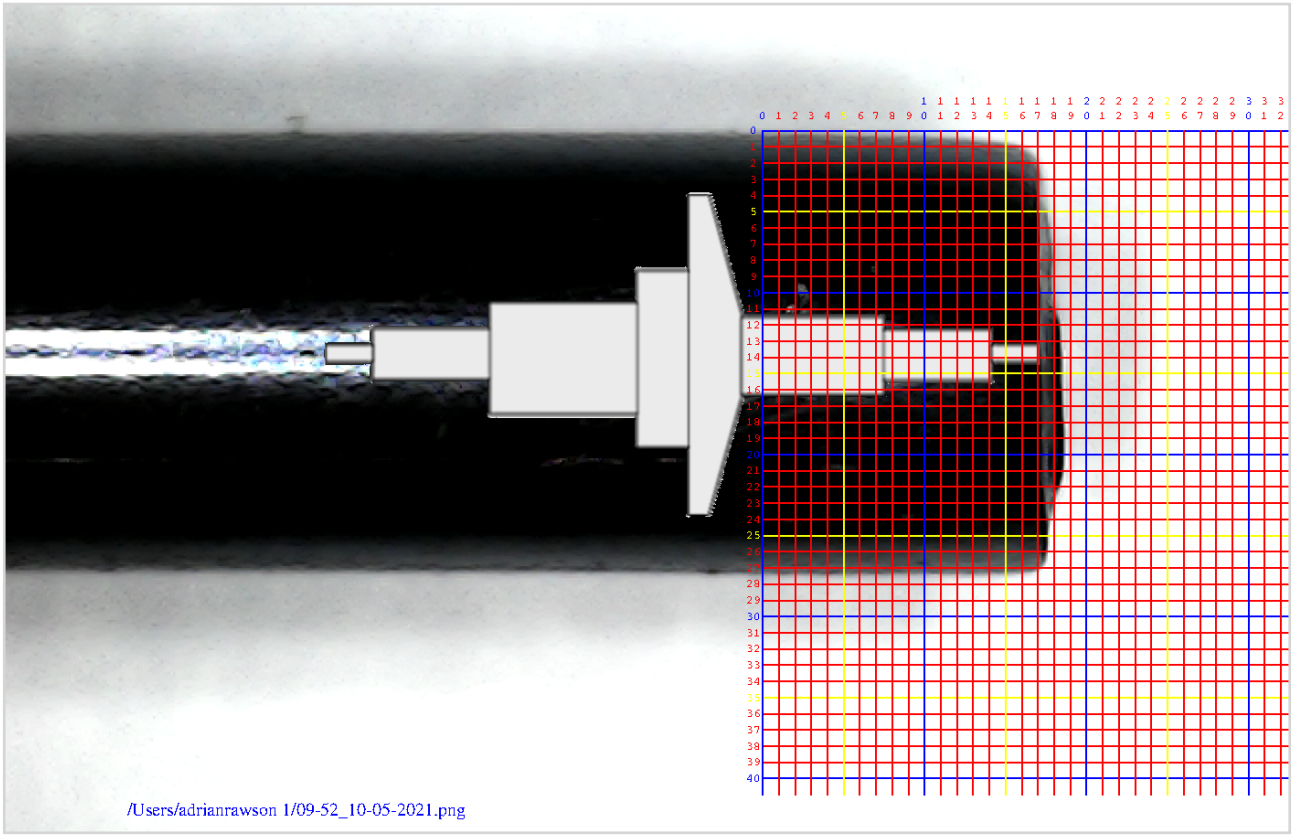
 Adrian Rawson © 2021

PHOTO

At the bottom right of the last graphic is a PHOTO button. To save a snapshot of the current screen left click this button.

The file name is displayed in blue while the photo is being generated followed by the word ‘DONE’. This can take many seconds on a Raspberry Pi so be patient.

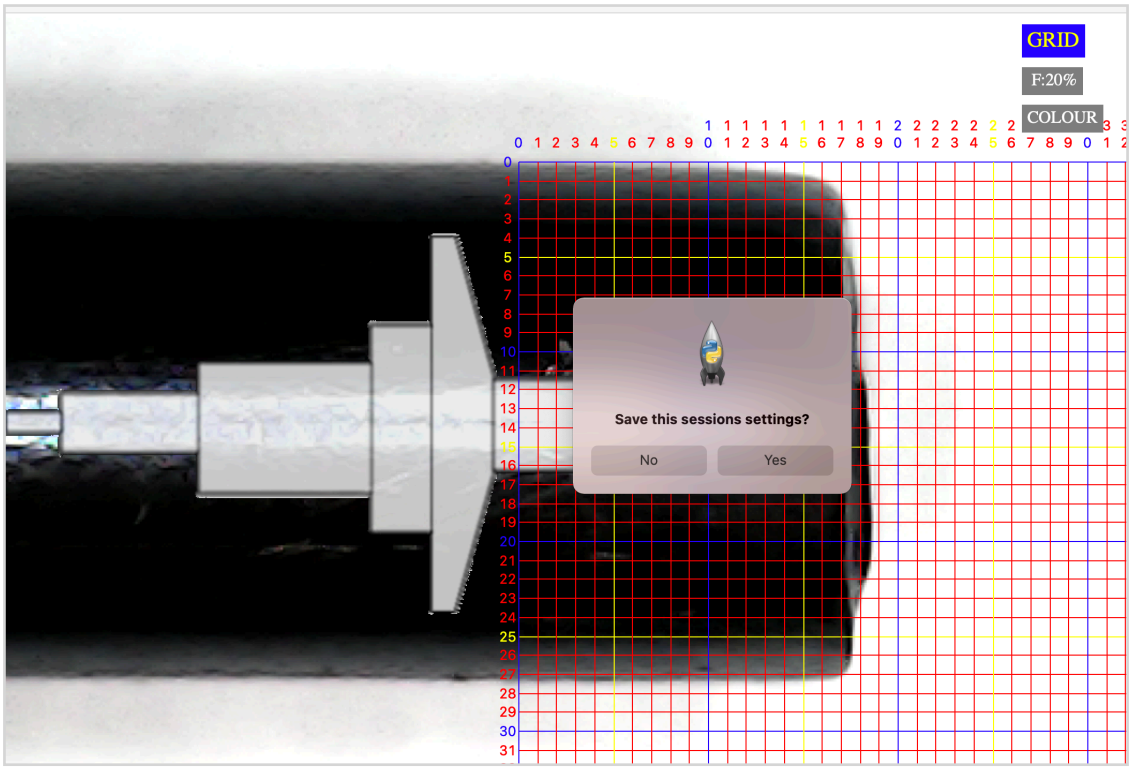
The ‘Postscript’ module used to generate the photo doesn’t support an alpha layer so unfortunately the transparency of the image is converted to a solid grey shade based on its fade value as shown below.



 Adrian Rawson © 2021

Closing and Saving

When the application is closed a '**Save this sessions settings?**' dialog is displayed.



Clicking on the '**Yes**' button will write the current state of the application to 'PiLoupe.dat'. When the application is run the existence of this file is used to configure its settings.

Clicking on the '**No**' button deletes the settings file whereupon it is regenerated with an initialised state the next time the application is run.

If a new image is to be loaded and used then '**No**' should be selected.

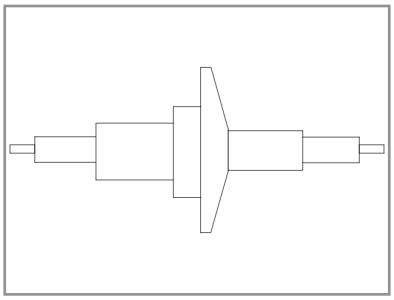
 Adrian Rawson © 2021

Creating Images

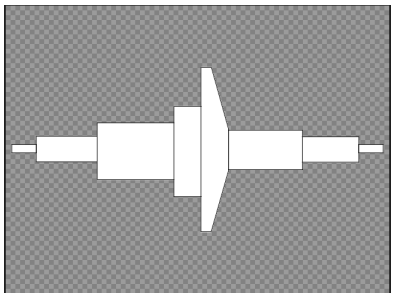
Any image can be used as long as it's smaller than the current screen size.

If the image doesn't have a transparent background then the whole image will be faded with the **F(ade)** button.

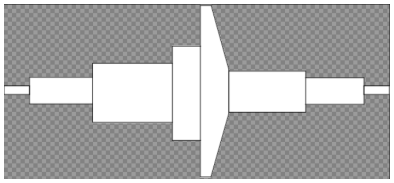
The images shown in this user guide were created using a CAD programme (QCAD on Mac in my case) and exported in 'png' format with a size of 640x480.



They were then loaded into a graphics editor (Pixelmator Pro on Mac in my case). Using the graphics editor, the background of the image was selected and deleted making it transparent.



The image was then cropped to remove as much of the transparent background as possible, and finally exported as the new '.png' image.



 Adrian Rawson © 2021