Automated translating beam profiler for in situ laser beam spot-size and focal position measurements

James Keaveney

Automated translating beam profiler for *in situ* laser beam spot-size and focal position measurements

James Keaveney[a)]

Joint Quantum Centre (JQC) Durham-Newcastle, Department of Physics, Durham University, South Road, Durham DH1 3LE, United Kingdom

We present a simple and convenient, high-resolution solution for automated laser-beam profiling with axial translation. The device is based on a Raspberry Pi computer, Pi Noir CMOS camera, stepper motor, and commercial translation stage. We also provide software to run the device. The CMOS sensor is sensitive over a large wavelength range between 300 and 1100 nm and can be translated over 25 mm along the beam axis. The sensor head can be reversed without changing its axial position, allowing for a quantitative estimate of beam overlap with counter-propagating laser beams. Although not limited to this application, the intended use for this device is the automated measurement of the focal position and spot-size of a Gaussian laser beam. We present example data of one such measurement to illustrate device performance. © 2018 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/). https://doi.org/10.1063/1.5022973

I. INTRODUCTION

There are a vast number of situations in experimental atomic and optical physics where precise measurement of laser beam profiles is required, from measurements of absolute beam intensity in determining atomic transition dipole matrix elements,[1] evaluating trap depths in optical dipole traps,[2] and laser beam shaping applications,[3] amongst others. For measuring the focal spot-size of a beam in one dimension, the traditional method is the knife-edge technique,[4–7] where the relative transmission is recorded as a sharp edge is moved through the cross section of the beam. The knife-edge method has a resolution that depends on the quality of the knife-edge and can be as low as 1 μm,[3] but finding the focal position using this method is a very time consuming and repetitive task, especially if more than one beam axis is to be measured. A labour-saving alternative is to use a CMOS image sensor to map the 2D spatial profile of the laser beam. This approach is widely used; there have been recent demonstrations using a webcam[8] or the camera built into a smart phone device for this application.[9]

However, for a focussed laser beam, one often needs to measure both the beam size at the focus and the axial position of the focus, which necessitates translating the camera or knife-edge along the axial dimension and repeating the profile measurements, which quickly becomes laborious if done manually. Commercial translating beam profilers are available for this purpose but are often prohibitively expensive (many thousands of US dollars) and relatively bulky. A recent novel approach used a spatial light modulator to negate the need for any physical translation of the beam,[10] but this was only demonstrated for relatively large (of order 1 mm) beam sizes.

Here we present a device based on a translating CMOS sensor head that uses a Raspberry Pi computer, the Pi Noir CMOS camera, and an inexpensive commercial translation stage. In addition to the hardware, we have developed a computer program to run the image acquisition and analysis.

Our design has two main advantages over current commercial alternatives: first, the small sensor head is the only part of the instrument that is located in the beam path—the main body of the device is relatively small (footprint 245 × 85 mm) and sits adjacent to the beam path, which facilitates *in situ* use of the device for most applications. Second, the sensor direction can be reversed without changing the axial position of the sensor (within machining tolerances), allowing for precise determination of the overlap between two beams, which finds use in many optical systems with overlapping counter-propagating beams, for example, in electromagnetically induced transparency[11,12] or four-wave mixing experiments.[1,13,14] Our device is also significantly less expensive than commercial alternatives that perform similar functions. In addition, the pixel size of the sensor is smaller than many current commercial systems, though this comes at the cost of a smaller total sensor area; this device is therefore more suitable in the measurement of relatively small beams. All of the computer software, hardware drawings, computer aided design (CAD) files, and electronics schematics including a bill-of-materials are available on the GitHub repository for the device.[15]

II. METHODS

A. Hardware

Figure 1 shows a rendering of the device. The camera is mounted on a plate and attached to the translation stage with standard opto-mechanical hardware (part numbers provided in the GitHub repository), allowing for coarse placement in the
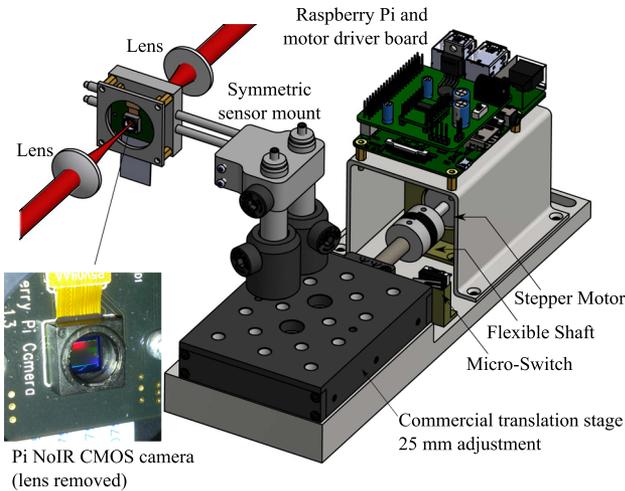
a)james.keaveney@durham.ac.uk

FIG. 1. Schematic illustration of the beam profiler assembly, showing a typical use scenario where the sensor head is placed between two short focal length lenses (50 mm in this case). See the main text for details of the highlighted components. Connecting cables are not shown for clarity. The photo shows the bare image sensor after the original lens has been removed.

xy plane. To allow for direct profiling of counter-propagating beams, for precise alignment of the relative focal planes, the camera can be placed facing either forwards or backwards along the translation axis. In either orientation, the mounts are designed symmetrically around the image-sensor plane, such that the sensor lies at the same axial coordinate, to within machinable precision. The camera attaches to the Raspberry pi (version 1 model B+ or version 2 or 3 model B) through a ribbon cable, allowing it to move freely (the standard cable is a little short for this application; several online retailers sell longer cables at little cost). A stepper motor controls the movement of the translation stage along the z-axis. The motor and stage are attached via a custom fine-adjustment screw and flexible shaft coupler, to allow for a small mismatch between the axes of the motor and screw thread. Finally, the whole assembly is mounted on a steel base plate with dimensions 245×85 mm. Control of the stepper motor is via a Pololu driver board based on the DRV8825 controller IC and is attached to the General Purpose Input/Output (GPIO) pins of the Raspberry Pi. The angular resolution of the stepper motor and the pitch of the screw thread set a limit to the axial precision of the device. In our case, with a 0.5 mm pitch screw thread and a stepper motor with 0.9° resolution (Nanotec ST4209S1006-B), the axial resolution limit is 1.25 μm. The maximum travel of the translation stage is 25 mm, and the step size can be set in software. When the profiler software starts, the zero-position is calibrated by moving the translation stage until a microswitch is actuated. In order to expose the bare sensor, the Pi NoIR CMOS camera has its lens removed (see the inset photograph). There are two versions of the PiNoIR camera: version 1 (pre-2016) uses the OmniVision OV5647 sensor with 2592×1944 pixels (pixel size 1.4×1.4 μm), whereas version 2 (April 2016 onwards) uses the Sony IMX219 sensor with 3280×2464 pixels (pixel size 1.12×1.12 μm). Both sensors share the same bit depth of 10-bits and physical sensor size of 3.67×2.74 mm. We have successfully tested the device with both sensors, and the software automatically detects which

sensor is connected. The smallest and largest beam sizes that can be detected are set by the total area of the sensor and the size of a single pixel. For the higher-resolution sensor, the minimum beam width which is practically possible to profile is around 2 μm, and the maximum practical beam waist is around 1.5 mm (for both sensors).

To use the device, the only requirements are a suitable power supply (8–12 V DC, capable of supplying > 2 A), a computer monitor attached to the High-Definition Multimedia Interface (HDMI) port of the Raspberry Pi, and a mouse and keyboard attached via universal serial bus (USB). All processing is done on the Raspberry Pi. A USB memory stick or similar can be used to extract the gathered data. All custom-made components have relatively simple form, amenable to manufacture in any reasonably equipped workshop.

B. Software

The beam profiler program can be set up to automatically start when the Raspberry Pi powers on. The analysis program has a graphical user interface and is written in Python. Installation instructions and the source code are provided on the main GitHub page for the beam profiler.[15]

For alignment purposes, the "Live View" mode can be enabled which uses the camera in the video mode. However, to avoid the automatic image enhancement techniques commonly used in digital photography which would artificially alter the data in an unknown way, the raw sensor data must be accessed. Since the sensor is a color camera, the raw pixels have color filters arranged in a Bayer pattern (alternating rows of red/green and green/blue pixels). Raw images are captured from the Bayer data of the sensor and processed into a 2D array. From this array, there are software options to interpolate the data back to the full resolution of the sensor (demosaic) or alternatively extract just the red-, green-, or blue-filtered pixels, which therefore increases the effective pixel separation by a factor of 2. In principle, this resolution decrease can be avoided without interpolation by use of a suitable correction factor,[8] but since this has to be empirically determined for each wavelength of light we have not implemented it here. Image exposure time can be adjusted in software, between a minimum of 9 or 12 μs (v1 and v2 sensors, respectively) and over 1 s. A background frame can be measured and subtracted from subsequent images. Once this is removed, the pixel response increases linearly with incident light intensity with near-zero offset, as shown in Fig. 2. The sensor behaves linearly until individual pixels start to saturate; therefore, the auto-exposure routine we provide in the software acquires images and adjusts exposure time until the maximum value of any single pixel on an image is just below saturation, to avoid any issues with non-linearity.

The user selects the range and resolution of the translation (z-axis) scan, and images are acquired at each position. After each image is acquired, the program integrates over each axis and fits Gaussian functions to the x and y axes separately, extracting the $1/e^2$ radii w_x and w_y. This assumes that for an elliptical beam the principal axes of the ellipse are aligned with the camera axes. For more complex beam analysis, the user can choose to automatically save the image data at each
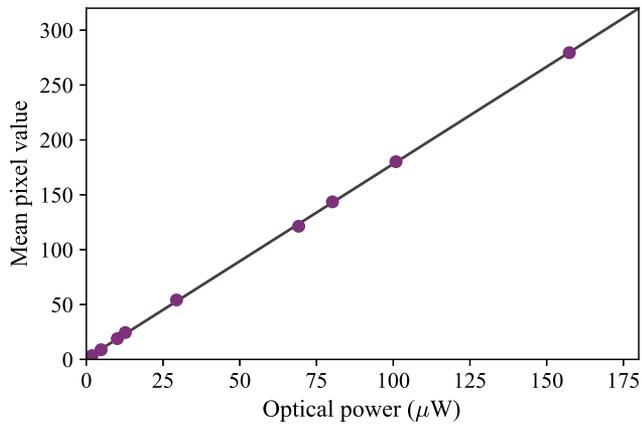
FIG. 2. Sensor linearity data for the version 2 sensor. Beam width approximately 2 mm and exposure time 85 μs. The fit is to a linear function with the intercept set to zero. The statistical error bars on the points are smaller than the data markers.

FIG. 3. Example image showing a tightly focussed beam with a relatively small spot size. Integrating over each dimension of the image separately, we extract and fit Gaussian profiles, shown on the bottom and right panels, with $1/e^2$ radii as shown in the figure. The purple points are the data; the dashed black lines are the best fit lines.

z position and manually analyze the data. With the model 3 Raspberry Pi, each data point takes around 30 s to move the translation stage, acquire, process, and save the image, so a full scan can be completed in approximately 20 min depending on the selected translation axis resolution. After the scan is complete, the axial widths $w_{x,y}(z)$ are fitted to the form of a focussed Gaussian beam given by

$$w(z) = w_0\sqrt{1 + (z - z_0)^2/z_R^2}, \qquad (1)$$

where w_0 is the beam waist, z_0 is the focal position, and $z_R = \pi w_0^2/\lambda$ is the Rayleigh range with λ being the wavelength of light. However, this assumes a beam-quality factor $M^2 = 1$ (i.e., a perfect TEM$_{00}$ mode) which is rarely the case experimentally, so in practice we let z_R be a floating parameter in addition to w_0 and z_0. If the beam profile is Gaussian, then M^2 can be extracted by assuming that $z_R = \pi w_0^2/M^2\lambda$.[16] It is possible to implement a full M^2 measurement for non-Gaussian beams using this system with suitable changes to the image analysis routine; this woud require using the ISO standard[17] second-moment $D4\sigma$ beam width measurement instead of simple Gaussian fitting. However, since this is beyond our typical use case, we have not implemented it here.

III. RESULTS

In the remainder of the paper, we illustrate the use of the beam profiler through example data.

A diode laser with 780 nm wavelength was coupled through a single-mode optical fiber. After the fiber, the beam was collimated to a $1/e^2$ radius of approximately 0.8 mm, and then incident on a single-element aspheric lens with 15 mm focal length. The damage threshold was not explicitly tested, but to avoid damaging the sensor, we kept the incident optical power low (<1 mW) using neutral density filters. To avoid saturation effects (i.e., the sensor output reaches its maximum value), the optical power should be set by coarsely placing the sensor near the focus of the beam and adjusting the incident intensity such that any single pixel is not saturated with
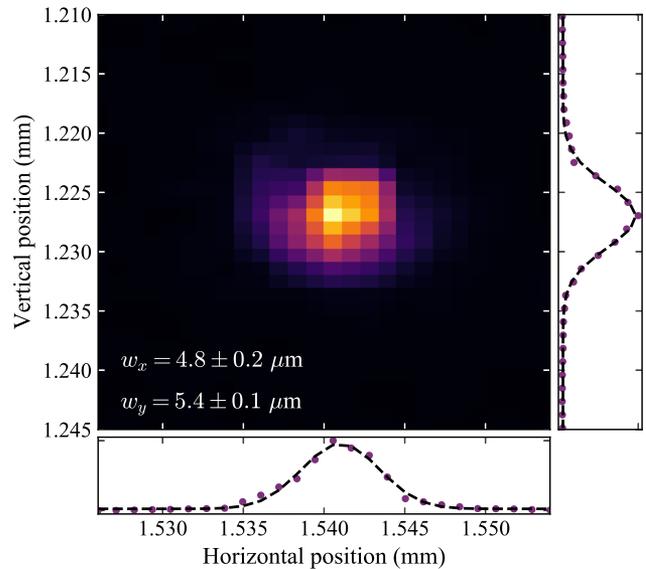
the lowest exposure time. Figure 3 shows an example image taken with the v2 sensor. The image is taken at the focal position of a 15 mm focal length aspheric lens, so the focal spot is only a few microns. We show the (cropped) image data and the integrated horizontal and vertical profiles, along with the fits to Gaussians. We extract radii $w_x = 4.8 \pm 0.2\ \mu$m and $w_y = 5.4 \pm 0.1\ \mu$m. These waists are consistent with 5.0 μm that we expect from simple Gaussian beam propagation[18] assuming a perfectly collimated input beam. The fit uses a standard Levenburg-Marquardt routine; the errors in individual data points are based on the square-root of the mean pixel value (Poisson statistics), and the errors in the fit parameters are estimated using the square-root of the diagonal elements of the covariance matrix.[19] In Fig. 4, we show an example of x- and y-axis beam waists as a function of translation distance. In this case, the selected axial resolution is 100 μm. After running a translation scan, the software fits the processed
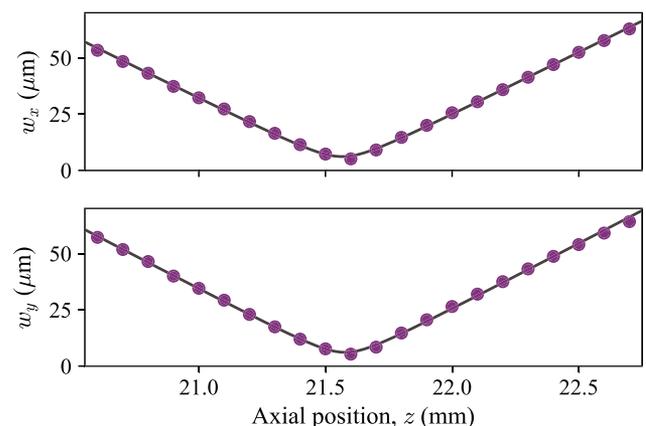

FIG. 4. Beam waists w_x and w_y as a function of axial position. Purple points are data; the statistical error bars from the individual fits to each image are smaller than the data point markers. The solid black lines are fits to Eq. (1).

beam width data to Eq. (1). The result of the fits is shown by the black lines; the x and y axes are fitted independently and both are clearly in agreement with the data. In this case, we find for the x-axis a focal size of $w_0 = 6.13 \pm 0.03$ μm, a focal position of $z_0 = 21.566 \pm 0.001$ mm, and a Rayleigh range of $z_R = 0.110 \pm 0.001$ mm. For the y-axis, we find $w_0 = 6.11 \pm 0.03$ μm, focal position $z_0 = 21.577 \pm 0.001$ mm, and Rayleigh range $z_R = 0.104 \pm 0.001$ mm. All error bars are taken from the square root of the diagonal elements of the covariance matrix. Assuming that $z_R = \pi w_0^2 / M^2 \lambda$, we can estimate the beam quality factor for the two axes as $M_x^2 = 1.38 \pm 0.01$ and $M_y^2 = 1.44 \pm 0.01$, which is reasonable for this type of laser.

IV. CONCLUSIONS

In conclusion, we have presented an automatic translating beam profiler that can be used for *in situ* measurements of focal spot size and positions. The system has some advantages over current commercial systems; the pixel size is smaller than many CCD-based commercial beam profilers, and the measurement head can be placed into an already existing optical setup, facing in both the $\pm z$ directions allowing for a quantitative estimate of beam overlap when using counter-propagating beams. The software is open-source and the analysis methods could be extended to include the ISO standard second-moment $D4\sigma$ beam width and M^2 measurements in future versions.

ACKNOWLEDGMENTS

APPENDIX: SOFTWARE AND DATA AVAILABILITY

In addition to the GitHub repository,[15] the data presented in this paper are available online at https://dx.doi.org/10.15128/r1dv13zt220.

[1] D. J. Whiting, J. Keaveney, C. S. Adams, and I. G. Hughes, Phys. Rev. A **93**, 043854 (2016); e-print arXiv:1602.08944.

[2] R. Grimm, M. Weidemüller, and Y. B. Ovchinnikov, Adv. At., Mol., Opt. Phys. **42**, 95 (2000); e-print arXiv:physics/9902072 [physics.atom-ph].

[3] *Laser Beam Shaping: Theory and Techniques*, 2nd ed., edited by F. M. Dickey (CRC Press, 2014).

[4] Y. Suzaki and A. Tachibana, Appl. Opt. **14**, 2809 (1975).

[5] J. M. Khosrofian and B. A. Garetz, Appl. Opt. **22**, 3406 (1983).

[6] W. Plass, R. Maestle, K. Wittig, A. Voss, and A. Giesen, Opt. Commun. **134**, 21 (1997).

[7] M. A. de Araújo, R. Silva, E. de Lima, D. P. Pereira, and P. C. de Oliveira, Appl. Opt. **48**, 393 (2009).

[8] G. Langer, A. Hochreiner, P. Burgholzer, and T. Berer, Opt. Lasers Eng. **51**, 571 (2013).

[9] M. A. Hossain, J. Canning, K. Cook, and A. Jamalipour, Opt. Lett. **40**, 5156 (2015).

[10] C. Schulze, D. Flamm, M. Duparré, and A. Forbes, Opt. Lett. **37**, 4687 (2012).

[11] J. Gea-Banacloche, Y.-Q. Li, S.-Z. Jin, and M. Xiao, Phys. Rev. A **51**, 576 (1995).

[12] H.-R. Noh and H. S. Moon, Phys. Rev. A **85**, 033817 (2012).

[13] Y.-S. Lee, S. M. Lee, H. Kim, and H. S. Moon, Opt. Express **24**, 28083 (2016); e-print arXiv:1609.02378.

[14] D. J. Whiting, N. Šibalić, J. Keaveney, C. S. Adams, and I. G. Hughes, Phys. Rev. Lett. **118**, 253601 (2017); e-print arXiv:1612.05467.

[15] See https://github.com/jameskeaveney/RPi-Beam-Profiler for the beam profiler software, electronics schematics and hardware design files.

[16] T. F. Johnston, Appl. Opt. **37**, 4840 (1998).

[17] ISO 11146-1:2005(en) and 11146-2:2005(en), "Lasers and laser-related equipment—Test methods for laser beam widths, divergence angles and beam propagation ratios—Parts 1 and 2," 2005.

[18] G. Brooker, *Modern Classical Optics* (OUP, Oxford, 2003).

[19] I. G. Hughes and T. P. A. Hase, *Measurements and Their Uncertainties: A Practical Guide to Modern Error Analysis* (OUP, Oxford, 2010).