RetroScope: An Open Framework for Motorizing and Digitizing Vintage Microscopes

Marvin Carstensen[0009−0007−6047−7264]⋆,
Justin Carstensen,
Niklas Hargarter,
Nils Porsche[0009−0005−3672−9316],
Simon Olberding[0009−0006−3405−5204], and
Marc Aubreville[0000−0002−5294−5247]

Flensburg University of Applied Sciences,
Kanzleistraße 91-93, 24943 Flensburg, Germany
{marvin.carstensen,justin.carstensen}@stud.hs-flensburg.de
{nils.porsche,niklas.hargarter}@hs-flensburg.de
{simon.olberding,marc.aubreville}@hs-flensburg.de

Abstract. Many optical microscopes remain mechanically and optically useful for decades, even when they no longer match the digital and automated workflows expected in modern laboratories or teaching environments. Replacing such instruments can be expensive and wasteful. Yet, they are of limited use for recent, more digital, workflows. This paper presents RetroScope, an open retrofit framework that motorizes and digitizes vintage optical microscopes while preserving the existing microscope body. The platform contributes a parameterized mechanical adapter strategy for coupling low-cost motors to existing focus and XY-stage controls, an embedded Raspberry Pi-based hardware and software stack, image-based calibration for pixel scale, stage scale, and backlash compensation. Autofocus, focus stacking, and tile scanning are implemented as integration examples that exercise calibrated motion and image capture. A prototype was evaluated on a vintage Nikon Optiphot 66 upright microscope using H&E-stained animal specimen and calibration slides. On the 4x objective, repeated stage-scale calibration produced 200-step standard deviations of 0.058 and 0.066 μm/step for X and Y. A hysteresis backlash model reduced the 200-step X forward error from 204.3 ± 10.8 μm without compensation to 11.4 ± 2.9 μm. These results support RetroScope as a resource-conscious, extensible prototype for accessible microscope automation, while also showing the limits of low-cost open-loop mechanics.

Keywords: Low-resource digital pathology · Microscope automation · Open hardware · Image-based calibration

⋆ Corresponding author: marvin.carstensen@stud.hs-flensburg.de

1 Introduction

Computational pathology relies on digitized microscope images that can be stored, inspected, shared, and processed. Whole-slide scanners and integrated motorized microscopes provide this capability in many modern workflows, but their cost and hardware assumptions can limit access in teaching, small laboratories, and low-resource environments. At the same time, many older optical microscopes remain mechanically and optically useful. Their limitation is often not the lenses or frame, but the absence of motorized control, digital capture, calibrated measurement, and programmable capturing.

Retrofit automation is therefore an attractive middle ground. Instead of replacing the microscope, a retrofit can attach motors and electronics to the manual controls that already exist. This is especially relevant for histology and pathology-adjacent education or in low-resource settings, where access to slide digitization can be more important than the absolute throughput of a commercial whole-slide scanner. The approach also connects to open scientific hardware, where shared designs and low-cost components can improve accessibility and reproducibility [15, 3].

The challenge is that a microscope retrofit is not merely a motor bracket. A useful platform must adapt to microscope-specific geometry, drive controls that were designed for control by human hands, account for open-loop stepper motion and backlash, capture images, store metadata, and expose workflows that are useful to an operator. Low-cost mechanics also make accuracy conditional: motor steps must be related to specimen-plane displacement through calibration, and motion compensation must be evaluated under realistic workflow conditions.

This paper presents RetroScope, an open platform for motorizing and digitizing vintage microscopes. It provides a unified, Apache 2.0-licensed framework that can be attached to a variety of non-motorized optical microscopes. The main contributions are:

1. A parameterized adapter for focus knobs, XY-stage knobs, camera ports, and mounting points.
2. A low-cost embedded hardware and software stack for live imaging, manual control, capture management, and automation.
3. Image-based calibration chain and backlash-compensation for making open-loop motion useful.

The implementation and design artifacts are available at:
https://github.com/DeepMicroscopy/RetroScope.

2 Related Work

Several open systems already address low-cost microscope motorization. Dantas de Oliveira et al. attach a servo-driven autofocus adapter and a separate motorized slide carrier to a conventional microscope. With an adjustable printed holders for a smartphone, the camera is used to capture images through the ocular of the microscope [7]. OpenStage couples stepper motors to micrometer-driven axes and demonstrates accurate open-loop stage motorization [5]. Schneidereit et al. provide a printable external positioning stage, electronics, and control code for automated high-content microscopy [18]. Lu et al. integrate open-loop micrometer stages, autofocus, scanning, and mosaicking into a complete low-cost microscope built from off-the-shelf optics rather than retaining an existing microscope body [12].

OpenFlexure Microscope demonstrated low-cost robotic microscopy using 3D-printed structures and open hardware [6]. Commercial microscope control systems and open software controllers such as Micro-Manager and Pycro-Manager enable flexible capturing when compatible motorized hardware is available [8, 17]. RetroScope does not introduce the general idea of low-cost microscope automation. Its distinction is the combination of a parameter-driven adapter to existing microscope controls with an integrated embedded application and image-based calibration chain.

3 RetroScope Framework

3.1 Reusable Microscope Interfaces

Most mechanical upright microscopes expose similar operator-facing interfaces: fine and coarse focus dials, stacked XY-stage knobs, a stable frame or base, illumination controls, and a camera or observation path. RetroScope uses these recurring interfaces as the abstraction layer for the retrofit. The platform does not assume access to the internal stage or focus mechanism. It only assumes that the existing controls for humans can be coupled, measured, and calibrated. Figure 1 shows how the printed interface modules, embedded electronics, and operator controls form the complete retrofit.

This distinction matters for universal adaptation. A single fixed printed part cannot fit various microscope bodies, because knob diameters, axis spacing, mounting points, and optical ports vary between different models and manufacturers. RetroScope therefore makes the interface configurable: the parametric model exposes focus-axis height, focus-axis radius, adapter offsets, and mounting-hole geometry as editable parameters. Adapting it begins by measuring the microscope base and feet, fine-focus shaft, stacked XY knobs, available clearance, and camera port. These measurements are entered into the Fusion model, the affected parts are regenerated and printed, and fit, axis travel, and collision clearance are checked before image-based calibration.

Parameter changes are intended to cover upright microscopes with the same external interface pattern: an accessible fine-focus shaft, externally driven XY-stage knobs, stable base mounting points, and a usable observation or camera path. Parameterization does not make the design universally compatible. A microscope without one of these interfaces, or with a substantially different stage or focus mechanism, requires a new interchangeable interface module and physical validation rather than only changed dimensions. The present work demonstrates

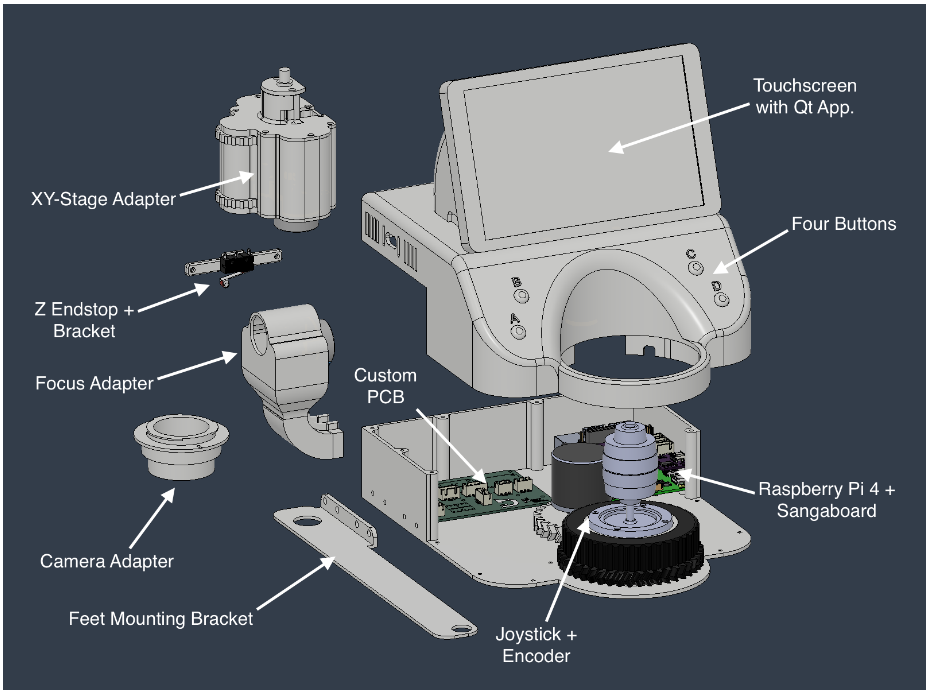


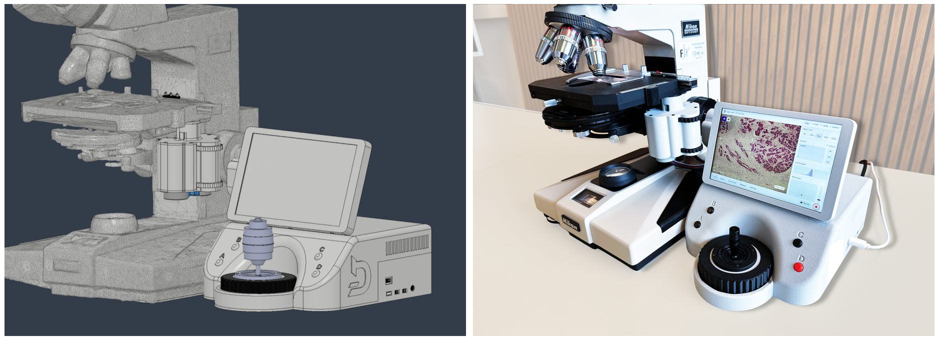
Fig. 1. Exploded render of the RetroScope prototype. The view separates the printed enclosure, joystick and encoder assembly, Raspberry Pi and Sangaboard electronics, custom PCB, feet mounting bracket, focus adapter, Z endstop bracket, XY-stage adapter, and camera adapter.

this procedure physically on one microscope and virtually through regenerated CAD variants.

3.2 Mechanical and Embedded Design

The prototype was built around a vintage upright microscope (Nikon Optiphot 66). The focus axis is driven by directly coupling a motor to the fine-focus axle through a printed shaft adapter. The coarse focus remains untouched. The XY stage uses a compact printed gearbox because the X and Y knobs are stacked and have different diameters. Printed gears are placed on top of the existing knobs. Motor gears and auxiliary manual knobs interface with them. The selected 12:24 tooth relation provides a 1:2 reduction, increasing torque and effective step resolution at the cost of additional gear backlash. Figure 2 compares the resulting CAD assembly with the fully assembled prototype attached to the baseline microscope.

The embedded platform combines a Raspberry Pi, touchscreen, camera capture device, Sangaboard stepper controller originally developed for the OpenFlexure ecosystem [2, 6], low-cost 28BYJ-48 stepper motors, joystick, rotary en-



(a) CAD render and microscope 3d scan. (b) Full assembled prototype.

Fig. 2. RetroScope retrofits an existing microscope rather than replacing it. The CAD model captures the adapter geometry and the physical prototype attaches motors, electronics, and touch display to the baseline microscope (Nikon Optiphot 66).

coder, endstop, and a custom PCB. The application is implemented in Python with PySide6/QML. It separates hardware drivers, services, domain logic, and QML bridges. The operator can use a live-view interface, joystick controls, gallery, calibration wizards, automation workflows, and a REST API for capture access and selected actions like autofocus. The touch interface shown in Fig. 3a exposes the camera preview together with the active objective, motion, and workflow controls. Captures are stored with OME-TIFF metadata for interoperability with microscopy tools [9, 11].

Prototype parts were printed in PETG for gears, buttons, and rapidly iterated parts and in ABS-GF for stiffer enclosure components. A 0.4-mm nozzle, 0.20-mm layers, 30% infill, and two perimeter walls were used.

Table 1 groups the priced bill of materials for the built prototype. The total excludes the existing microscope, DSLR camera, printer, and assembly tools. Prices depend on supplier and date. The table therefore documents affordability and reproducibility without claiming a universal build price.

Table 1. Categorized bill of materials for the evaluated prototype.

Category	Included components	Subtotal
Compute, display, capture interface	Raspberry Pi 4, DSI touchscreen, Cam Link 4K	EUR 282.88
Motion and mechanical control	Sangaboard, three motors, bearing, endstop	EUR 50.75
Inputs, electronics, and PCB	Joystick, ADS1115, encoder, INA219 modules, level converter, PCB, wiring and connectors	EUR 67.55
Print material	PETG and ABS-GF consumption	EUR 18.29
Total system cost	Excluding microscope, DSLR camera, fabrication, and tools	**EUR 419.47**

3.3 Calibration and Motion Compensation

RetroScope stores objective profiles with pixel scale, stage scale, numerical aperture, focus parameters, and backlash values. Pixel-scale calibration maps image pixels to specimen-plane micrometers. Stage-scale calibration maps motor steps to X/Y displacement. These two mappings are required for measurement and tile planning because image overlap must be translated into motor steps.

The system remains open-loop: it estimates position from motor commands, not from measured stage encoders. This makes backlash compensation central. Low-cost geared steppers are attractive for accessible hardware but introduce mechanical slack and other open-loop limitations [1]. Backlash and dead-zone compensation are established approaches for reducing predictable lost motion in mechanical control systems [13]. RetroScope applies this through a per-axis slack-state implementation. For axis i, a configured backlash width b_i and slack estimate s_i are maintained with

$$-\frac{b_i}{2} \leq s_i \leq \frac{b_i}{2}$$

For a compensated requested move d_i, the controller computes the slack take-up p_i as

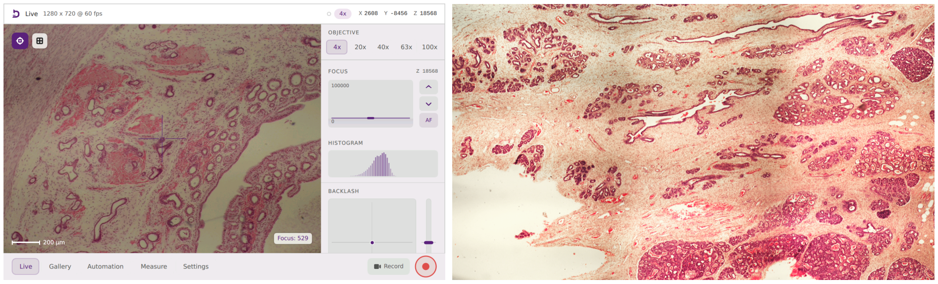
$$p_i = \mathrm{sign}(d_i)\frac{b_i}{2} - s_i$$

and sends a single motor command m_i as

$$m_i = d_i + p_i$$

After the command is accepted, the slack estimate is set to the band edge in the commanded direction. Raw moves can still update the estimate without applying compensation. The OpenFlexure software instead corrects an opposing-direction move by overshooting and returning so that the target is approached from a preferred direction [14]. RetroScope's stateful implementation computes only the still-untraversed part of the band and combines it with the requested displacement in one command. This supports arbitrary joystick and workflow move sequences, but it retains the usual limitations of model-based open-loop compensation.

The X/Y backlash widths are determined with an operator-assisted calibration workflow. After pre-moving an axis in one direction, the operator stores a central reference crop, reverses the axis in known increments, and compares the live crop with the reference. The cumulative reverse steps needed to re-establish visible motion are accepted as the mechanical backlash width and shared across objective profiles. Stage-scale is measured by phase correlation, with central template matching used as a confidence cross-check. Phase-correlation registration is established for subpixel displacement measurement [10].



(a) The live-view interface. (b) Stitched tile scan.

Fig. 3. RetroScope interface and workflow output. The live view provides camera preview, objective state, and motion controls, while the stitched tile scan illustrates how calibrated XY motion and image capture are combined to validate automated capturing.

4 Workflow Validation

RetroScope implements autofocus, focus stacking, and tile scanning as validation workflows since each one of these standard algorithms exercises multiple layers of the platform and is thus well-suited for this purpose.

Autofocus moves the Z axis through a configured search range and evaluates a focus score from the camera stream. The score is computed on grayscale/luma data from a central image region. It uses contrast-gated tiles, a light Gaussian blur to reduce sensor noise, variance-of-Laplacian [16], and Scharr-gradients. The strongest tile scores are pooled so that blank regions in the field of view do not dominate the focus estimate. The final focus position is refined from the highest-scoring samples by parabolic interpolation. This validates Z motion, focus scoring, and objective-specific parameters.

Focus stacking captures a sequence of Z planes and produces a blended result while preserving source images and metadata. Tile scanning captures adjacent fields of view as either a stitched panorama or a deterministic grid mosaic. The stitched result in Fig. 3b illustrates how calibrated XY motion, camera capture, and tile assembly are combined in one workflow. These captures are stored with objective, position, dimension, and workflow metadata. Dark tile seams remain visible in some outputs, likely due to illumination fall-off, vignetting, or other optical-path effects. Reducing this would require flat-field correction [4].

5 Evaluation

The evaluation used the complete prototype attached to the Nikon Optiphot 66: printed adapters, embedded electronics, custom PCB, Sangaboard-controlled motors, motion controller, Nikon D5100 SLR camera, and calibration workflows. A Nikon E Plan 4x/0.1 160/- objective and 1280x720-pixel stream through an Elgato Cam Link 4K were used. Calibration slides supported displacement measurements, while exemplary formalin-fixed, paraffin-embedded tissue stained with hematoxylin and eosin standard dye was used to demonstrate image capture. The experiments evaluate platform mechanics and calibration, not clinical diagnostic performance.

Adaptability was assessed as design-artifact evidence rather than as a second hardware experiment. Editing the microscope-specific Fusion parameters regenerated the mounting, focus, and XY assemblies for virtual body variants without rebuilding the model. This checks parameter propagation and geometric editability, but it does not establish physical fit, stiffness, or calibrated motion on another microscope. Those claims require a second microscope and prototype.

Stage-scale repeatability measured how consistently a fixed motor-step command maps to optical specimen displacement. The stage was moved independently on X and Y without compensation, and image shift was measured along the commanded axis. Pixel displacement was converted with the calibrated pixel scale and divided by 100 or 200 motor steps. Ten valid trials were recorded for every axis and command size. The 200-step condition was expected to be more informative because it traverses beyond the configured backlash widths of 100 X steps and 130 Y steps.

Motion accuracy used a forward-and-return sequence. A reference image was acquired, the axis was commanded forward by 100 or 200 requested stage steps, and a second image measured forward displacement. The same number of requested steps was then commanded in reverse, and a final image measured the two-dimensional residual from the starting view. Forward error is measured optical displacement minus the displacement predicted from the calibrated stage scale. Negative values indicate under-travel and positive values indicate over-travel. Reversal residual is the magnitude of the final image offset and is therefore non-negative. Ten valid trials compared raw commands, a simple sign-based reversal correction, and the slack-state compensation.

Calibration repeatability independently reran stage-scale and uncompensated reversal-residual measurements. It asks whether calibration produces similar profile values over repeated trials, not whether the open-loop stage returns to an absolute physical coordinate. Phase-correlation response and template matching were recorded for trial validation. Tables 2 and 3 report the retained valid measurements as mean $\pm$ sample standard deviation.

5.1 Motion and Calibration Results

Table 2 shows that separate axis calibration is necessary: the 200-step estimates were 2.223 μm/step for X and 3.136 μm/step for Y. Their standard deviations, 0.058 and 0.066 μm/step, were lower than the corresponding 100-step values. The repeated calibration series recovered similar 4x stage scales.

Table 3 exposes the axis- and distance-dependent behavior. At 200 steps, slack-state compensation reduced the magnitude of X forward error by 94.4%, from 204.3 to 11.4 μm, and reduced its reversal residual from 0.9 to 0.3 μm. For Y, it reduced forward-error magnitude by 86.2%, from 260.1 to 35.8 μm. Sign compensation had especially high variance on Y.

Table 2. 4x stage-scale and calibration-repeatability results.

Experiment	Axis/Condition	Unit	Result	n
Stage scale	X, 100 steps	μm/step	2.183 ± 0.095	10
Stage scale	X, 200 steps	μm/step	2.223 ± 0.058	10
Stage scale	Y, 100 steps	μm/step	3.410 ± 0.238	10
Stage scale	Y, 200 steps	μm/step	3.136 ± 0.066	10
Calibration repeat	X stage scale	μm/step	2.232 ± 0.070	10
Calibration repeat	Y stage scale	μm/step	3.148 ± 0.110	10
Calibration repeat	X reversal residual	μm	2.150 ± 1.385	10
Calibration repeat	Y reversal residual	μm	1.611 ± 1.464	10

Table 3. 4x motion results from ten trials per condition. Errors/Residuals are in μm.

Axis	Steps	Mode	Forward error	Reversal residual
X	100	Raw	-165.4 ± 68.7	40.0 ± 78.7
X	100	Sign	-61.7 ± 102.2	119.2 ± 109.6
X	100	Slack state	63.2 ± 56.0	41.5 ± 45.1
X	200	Raw	-204.3 ± 10.8	0.9 ± 0.7
X	200	Sign	-30.4 ± 58.5	44.8 ± 55.5
X	200	Slack state	11.4 ± 2.9	0.3 ± 0.2
Y	100	Raw	-264.7 ± 33.6	48.4 ± 104.5
Y	100	Sign	-186.9 ± 74.7	229.4 ± 96.1
Y	100	Slack state	165.5 ± 64.0	51.6 ± 49.4
Y	200	Raw	-260.1 ± 18.5	2.1 ± 2.8
Y	200	Sign	-60.6 ± 83.0	90.4 ± 107.4
Y	200	Slack state	-35.8 ± 16.4	52.6 ± 1.7

The 100-step results are substantially less stable. A 100-step request equals the configured X backlash width and is smaller than the 130-step Y width, so the modeled take-up is comparable to the requested displacement and small differences in initial engagement can dominate the result. The Y gearbox also has a longer unsupported printed path between the motor and microscope knob than X, making flex a plausible contributor to its directional response. This mechanical interpretation was observed during development but was not isolated experimentally. Friction and print tolerance may also contribute.

6 Discussion and Conclusion

The results support RetroScope as a working low-cost retrofit prototype for microscope automation. The framework utilizes a parameterizable mechanical adapter, which allows it to be fitted to most mechanical microscopes, with embedded control hardware, modular software, image-based calibration, and workflow automation. Relative to previous retrofit systems, RetroScope's distinction is the combination of a parameter-driven adapter to existing microscope controls with an integrated embedded application and image-based calibration chain.

Autofocus, focus stacking, tile stitching, phase correlation, and motion compensation are established techniques used to exercise this platform, not claimed as new algorithms.

For low-resource computational pathology, RetroScope offers a route for bringing serviceable microscopes into digital capturing workflows without replacing the original instrument and may extend the functional lifespan of equipment in teaching environments and laboratories.

The evaluation also bounds the claims. RetroScope is not a precision stage and does not replace a clinical whole-slide scanner. It controls a chain of existing microscope mechanics, printed gears, low-cost steppers, and open-loop estimates. Backlash compensation can improved predictable 200-step forward lost motion, especially on X, but both 100-step conditions show that compensation cannot remove flex, friction, missed steps, or manual disturbances. Quantitative motion evaluation used only a 4x objective. Performance at pathology-relevant 20x and 40x magnification, where fields of view and focus tolerances are smaller, has not been established.

Several limitations therefore remain. The prototype was validated on a single microscope body, so the parametric model demonstrates adaptability as a procedure and through virtual regeneration, not as broad empirical compatibility across microscope families. The system tracks commanded motor position rather than measured sample position, meaning calibration can become invalid after manual movement, loosened parts, changed friction, or missed steps. This paper also does not quantify tile-alignment accuracy, overlap registration error, structural similarity (SSIM), focus-stack quality, optical resolution, or diagnostic image quality. The workflow images demonstrate integration but cannot establish suitability for clinical use.

Nevertheless, the prototype demonstrates that vintage microscopes can be extended into programmable digital imaging systems without discarding the original optical instrument. The same interface strategy could be extended for other focus mechanisms, XY-control layouts, camera ports, or motor controllers. The service architecture and REST interface could similarly support time-lapse acquisition, remote experiment clients, and downstream computational analysis, but these extensions have not been implemented or evaluated here. Future work should prioritize physical fitting on additional microscope bodies, 20x/40x calibration and motion tests and closed-loop sample-position sensing.

Acknowledgments. N.P. and M.A. acknowledge support by Deutsche Forschungsgemeinschaft (DFG), project OBELISK (545049923).

Disclosure of Interests. The authors have no relevant financial or non-financial interests to disclose.

References

1. Acarnley, P.P.: Stepping Motors: A Guide to Theory and Practice. Institution of Electrical Engineers, London, 4 edn. (2002)
2. Ayazi, F.: Sangaboard RP2040 hardware design. GitLab repository (2022), https://gitlab.com/filipayazi/sangaboard-rp2040, accessed 2 August 2026
3. Baden, T., Chagas, A.M., Gage, G., Marzullo, T., Prieto-Godino, L.L., Euler, T.: Open labware: 3-d printing your own lab equipment. PLOS Biology **13**(3), e1002086 (2015). https://doi.org/10.1371/journal.pbio.1002086
4. Bowman, R.W., Vodenicharski, B., Collins, J.T., Stirling, J.: Flat-field and colour correction for the raspberry pi camera module. Journal of Open Hardware **4**(1) (2020). https://doi.org/10.5334/joh.20
5. Campbell, R.A.A., Eifert, R.W., Turner, G.C.: OpenStage: A low-cost motorized microscope stage with sub-micron positioning accuracy. PLOS ONE **9**(2), e88977 (2014). https://doi.org/10.1371/journal.pone.0088977
6. Collins, J.T., Knapper, J., Stirling, J., et al.: Robotic microscopy for everyone: The OpenFlexure Microscope. Biomedical Optics Express **11**(5), 2447–2460 (2020). https://doi.org/10.1364/BOE.385729
7. Dantas de Oliveira, A., Rubio Maturana, C., Zarzuela Serrat, F., et al.: Development of a low-cost robotized 3D-prototype for automated optical microscopy diagnosis: An open-source system. PLOS ONE **19**(6), e0304085 (2024). https://doi.org/10.1371/journal.pone.0304085
8. Edelstein, A.D., Tsuchida, M.A., Amodaj, N., Pinkard, H., Vale, R.D., Stuurman, N.: Advanced methods of microscope control using Micro-Manager software. Journal of Biological Methods **1**(2), e10 (2014). https://doi.org/10.14440/jbm.2014.36
9. Goldberg, I.G., Allan, C., Burel, J.M., et al.: The Open Microscopy Environment (OME) data model and XML file: Open tools for informatics and quantitative analysis in biological imaging. Genome Biology **6**(5), R47 (2005). https://doi.org/10.1186/gb-2005-6-5-r47
10. Guizar-Sicairos, M., Thurman, S.T., Fienup, J.R.: Efficient subpixel image registration algorithms. Optics Letters **33**(2), 156–158 (2008). https://doi.org/10.1364/OL.33.000156
11. Linkert, M., Rueden, C.T., Allan, C., et al.: Metadata matters: Access to image data in the real world. Journal of Cell Biology **189**(5), 777–782 (2010). https://doi.org/10.1083/jcb.201004104
12. Lu, Q., Liu, G., Xiao, C., et al.: A modular, open-source, slide-scanning microscope for diagnostic applications in resource-constrained settings. PLOS ONE **13**(3), e0194063 (2018). https://doi.org/10.1371/journal.pone.0194063
13. Nordin, M., Gutman, P.O.: Controlling mechanical systems with backlash—a survey. Automatica **38**(10), 1633–1649 (2002). https://doi.org/10.1016/S0005-1098(02)00047-X
14. OpenFlexure Project: Sangaboard microscope-stage documentation, https://openflexure-microscope-software.readthedocs.io/, accessed 2 August 2026
15. Pearce, J.M.: Building research equipment with free, open-source hardware. Science **337**(6100), 1303–1304 (2012). https://doi.org/10.1126/science.1228183
16. Pech-Pacheco, J.L., Cristobal, G., Chamorro-Martinez, J., Fernandez-Valdivia, J.: Diatom autofocusing in brightfield microscopy: A comparative study. In: Proceedings of the 15th International Conference on Pattern Recognition. vol. 3, pp. 314–317 (2000). https://doi.org/10.1109/ICPR.2000.903548

17. Pinkard, H., Stuurman, N., Ivanov, I.E., et al.: Pycro-Manager: Open-source software for customized and reproducible microscope control. Nature Methods **18**, 226–228 (2021). https://doi.org/10.1038/s41592-021-01087-6
18. Schneidereit, D., Kraus, L., Meier, J.C., Friedrich, O., Gilbert, D.F.: Step-by-step guide to building an inexpensive 3D printed motorized positioning stage for automated high-content screening microscopy. Biosensors and Bioelectronics **92**, 472–481 (2017). https://doi.org/10.1016/j.bios.2016.10.078