



Kent Academic Repository

Hughes, Michael (2023) *Real-timing processing of fiber bundle endomicroscopy images in Python using PyFibreBundle*. Applied Optics . ISSN 0003-6935.

Downloaded from

<https://kar.kent.ac.uk/102509/> The University of Kent's Academic Repository KAR

The version of record is available from

<https://doi.org/10.1364/AO.503700>

This document version

Publisher pdf

DOI for this version

Licence for this version

CC BY (Attribution)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in **Title of Journal**, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).



Real-time processing of fiber bundle endomicroscopy images in Python using PyFibreBundle

MICHAEL R. HUGHES 

Applied Optics Group, School of Physics and Astronomy, University of Kent, Canterbury, CT2 7NH, UK (m.r.hughes@kent.ac.uk)

Received 23 August 2023; revised 30 October 2023; accepted 30 October 2023; posted 31 October 2023; published 21 November 2023

Fiber imaging bundles allow the transfer of optical images from place-to-place along narrow and flexible conduits. Traditionally used extensively in medical endoscopy, bundles are now finding new applications in endoscopic microscopy and other emerging techniques. PyFibreBundle is an open-source Python package for fast processing of images acquired through imaging bundles. This includes detection and removal of the fiber core pattern by filtering or interpolation, and application of background and flat-field corrections. It also allows images to be stitched together to create mosaics and resolution to be improved by combining multiple shifted images. This paper describes the technical implementation of PyFibreBundle and provides example results from three endomicroscopy imaging systems: color transmission, monochrome transmission, and confocal fluorescence. This allows various processing options to be compared quantitatively and qualitatively, and benchmarking demonstrates that PyFibreBundle can achieve state-of-the-art performance in an open-source package. The paper demonstrates core removal by interpolation and mosaicing at over 100 fps, real-time multi-frame resolution enhancement and the first demonstration of real-time endomicroscopy image processing, including core removal, on a Raspberry Pi single board computer. This demonstrates that PyFibreBundle is potentially a valuable tool for the development of low-cost, high-performance fiber bundle imaging systems.

Published by Optica Publishing Group under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

<https://doi.org/10.1364/AO.503700>

1. INTRODUCTION

Fiber imaging bundles are thin, flexible conduits that can be used to transfer images from locations that are otherwise difficult to access with cameras and imaging systems. The bundles contain tens of thousands of closely packed fiber cores, with the relative positions of the cores preserved along the length of the bundle. This “coherent” arrangement of the fiber cores distinguishes imaging bundles from “incoherent” bundles that can only be used for illumination. When a light field is projected onto one end of an imaging bundle, each fiber core effectively collects and transmits one intensity value or “pixel,” resulting in a pixelated reconstruction of the light field at the far end.

Fiberscopy (flexible medical endoscopy) has relied on fiber imaging bundles since its development in the 1950s, originally with eye-pieces for the operator to look directly at the transmitted image, and then later with cameras built into endoscope handle [1]. More recently, the development of high-quality chip-on-tip cameras has led to imaging bundles falling out of favor for video endoscopy in many countries, although it is still used in some parts of the world as well as for specialized applications requiring ultra-thin endoscopes. However, new

medical applications of imaging bundles have been found such as in endoscopic microscopy. A fiber bundle with a bare tip, or with the addition of a micro-lens, can be used as the probe of a video microscope, offering an *in vivo* alternative to traditional biopsy and histology for high-resolution tissue imaging.

There are two principal types of bundles used for imaging: fused and leached. In fused bundles, the fiber cores are embedded in a single shared cladding, while in leached bundles each core has its own independent cladding, separated from the other fibers by a spacer material. The spacer material is removed along most of the length of the bundle as part of the leaching process during manufacture, resulting in a much smaller bending radius compared with fused bundles, in exchange for a generally lower core density. Both types of fiber bundles can be made small and flexible enough (around 1 mm or smaller) to be deployed through working channels of endoscopes. This allows endoscopic microscopy to be combined with conventional endoscopy for steering and guidance.

Fiber bundles are particularly advantageous for *in vivo* fluorescence confocal microscopy, where a laser beam is scanned in a raster pattern over the tissue and the fluorescence returning

from each point is imaged through a pinhole. While miniaturized scanning systems suitable for endoscopy have been widely demonstrated (see for example [2,3]), a simpler solution is to scan a laser over the proximal end of a fiber bundle outside of the patient and, thus, have the scanning pattern transferred to the tissue without any need for physical scanning at the distal tip of the probe. Confocal microscopy through fiber bundles was demonstrated by Gmitro and Aziz in 1993 [4] and has been explored extensively since, particularly using the Cellvizio endomicroscope from Mauna Kea Technologies [5]. While limitations include the coupling efficiency and level of autofluorescence generated within the different types of fiber bundles, which ultimately influences the signal-to-noise ratio of the images, confocal fluorescence endomicroscopy has been successfully investigated for a wide range of clinical applications, particularly in the gastrointestinal tract [6], using both intravenous fluorescein and topical fluorescent stains.

Many other variations of fiber bundle endomicroscopy have also been demonstrated, both pre-clinically and clinically. A non-sectioning widefield endomicroscope can easily be built using a fiber bundle [7], and with a careful choice of topical fluorescent contrast agents this offers an effective and low-cost approach to *in vivo* microscopy for applications such as detecting oral cancer [8]. Line-scanning [9,10] and structured-illumination endomicroscopes [11–13] trade off partial optical sectioning in exchange for higher frame rates. Other techniques include multi-spectral imaging [14], two photon microscopy [15], fluorescence lifetime imaging [16], holographic microscopy [17,18], light field imaging [19], and optical coherence tomography [20]. White light imaging is also possible, in both confocal [21] and non-confocal [22] modes, and by using multiple illumination sources it is possible to obtain additional contrast through phase [23]. A considerable body of work has also explored using phase conjugation techniques for more precise control of a scanning laser spot at the distal end (see [24] for a review), although these approaches have yet to find clinical applications.

A. Image Processing of Fiber Bundle Images

To obtain good quality images from fiber bundle systems, particularly for microscopy, a number of image processing steps must be performed. Methods for performing this processing have been the focus of extensive research over recent years, and a comprehensive review is available in [25]. Only a selection of relevant research is reviewed here.

The raw images captured through fiber bundles suffer from many artifacts. Most obviously, the individual fiber cores are visible, with dark spaces corresponding to the cladding in between; this is often known as the “honeycomb effect.” In fused bundles, such as those produced by Fujikura (Japan), each core tends to have a slightly different size and shape; this is by design to minimize cross-core coupling, but it leads to each core having a different coupling and transmission efficiency. In fluorescence applications, a background signal may be generated by the fiber cores, leading to an offset in the image. The first step in processing images is, therefore, to remove any background, normalize the transmission factors between the different cores, and then remove the honeycomb pattern.

A wide range of methods for removing the honeycomb core pattern have been reported, and these are summarized in [25]. A broad class of methods involves convolution with a smoothing kernel; approaches range from simple generic low-pass filters (such as Gaussian [15]) to filters designed specifically for core removal [26,27]. Early in the development of endomicroscopy, the authors of [28] introduced an approach based on triangular linear interpolation, in which the intensity of each core is extracted and then interpolated onto a Cartesian grid. While this method may superficially appear computationally expensive, it is amenable to pre-computation and so is practical for real-time use. It has also been extended to color imaging [29]. A number of more complex approaches have also been suggested, including iterative threshold shrinking using L1-minimization [30] and compressive sensing [31]. Approaches that involve iterative steps have less practical utility, at least for online use, and most published systems continue to use variations of simple spatial filtering and interpolation.

A limitation of fiber bundle imaging in general is the limited number of cores in the bundle and the spacing between them. For a fiber bundle without a lens at the distal end, the resolution is typically around twice the core spacing. Adding a distal lens with non-unity magnification improves the resolution, but at the expense of reducing the field-of-view by the same factor. In effect, the number of cores in the bundle defines the number of resolution elements in the image, and the 30,000 cores of a typical fiber bundle compares poorly with the millions of pixels found on modern camera chips.

There are two general approaches for mitigating this problem of low pixel count: mosaicing and resolution enhancement. In mosaicing, images are stitched together as the probe is moved across the tissue, increasing the effective field-of-view. This can be done in real-time by only registering adjacent image frames and assuming that there is no non-rigid deformation of the tissue as the probe moves [32,33]. This approach tends to accumulate errors and does not handle the inevitable deformation of the tissue as the probe is dragged across it. More sophisticated algorithms have, therefore, been developed (e.g., [34]), which attempt to achieve a globally consistent registration, but these more complex approaches are better suited to post-processing rather than online use. The frame rate of the endomicroscope becomes a limiting factor in allowing mosaicing since there must be sufficient overlap between image frames, motivating the development of high-speed endomicroscopes for mosaicing [35], particularly when the probe is moved by mechanical means [36].

Resolution enhancement is possible if multiple images are captured with the sample slightly shifted with respect to the bundle [37–40]. The concept is similar to, but distinct from, pixel shifting approaches for improving resolution in camera images [41]. For each image, the tissue is sampled at a different location by the fiber bundle pattern, and so combining these images can improve the resolution. This works particularly well because of the low fill-factor of the bundle (i.e., there are dead spaces between the cores), and an improvement in resolution by a factor of 2 [39] can be achieved without relying on iterative algorithms conventionally used for pixel super-resolution. Rotation, as opposed to translation, can also be used to achieve some resolution enhancement [42,43]. Deep learning

approaches to resolution enhancement have also shown promise (e.g., [44,45]) but are not discussed further here.

B. Software for Image Processing

There is currently no standard open-source package for performing the full range of processing steps above. Some open source code is available, for example in C++ and MATLAB [46,47], but it only provides a subset of functionality and does not have full documentation. Academic papers use a variety of approaches, from displaying simply the raw images to variations on the standard methods described above, but often with incomplete descriptions of how the processing was performed and almost always no source code available. PyFibreBundle fills this gap by providing an easy-to-use and comprehensive set of fiber bundle imaging processing function, which can be used for real-time imaging. This will allow research teams developing new fiber bundle imaging techniques to more quickly develop software to test and validate their systems. As demonstrated below, it is fast enough to be used with low-end and single-board computers, opening up new opportunities to develop low-cost and portable fiber bundle imaging systems.

2. METHODS

The core functionality of PyFibreBundle was investigated using purpose-collected images from several fiber bundle imaging systems and used PyFibreBundle Release 1.3.4. Processing was performed on a desktop PC (Intel Core i7, quad core 12 GB RAM), a laptop (Intel Core i5 8th Gen, 8 GB RAM), and a Raspberry Pi 4 (Rev 2). Benchmarking on these three systems is presented in Section 3.E. A technical description of the parts of PyFibreBundle that are examined here is provided below, and the imaging systems used to generate images for testing are described in Section 2.J.

A. Overview of PyFibreBundle

PyFibreBundle is an open-source package written entirely in Python. It works with both monochrome and color images in a variety of data representations. As an interpreted language, Python is generally considered slow in comparison to compiled languages such as C++. However, this is balanced by its easy-of-use and widespread adoption throughout academia and industry, particularly in fields such as data science and machine learning. The slow speed of interpreted code can be mitigated through the use of compiled libraries, such as Numpy, Scipy, and OpenCV, to perform computationally intensive operations. Additionally, PyFibreBundle also uses the just-in-time compiler (JIT) Numba to accelerate small portions of code that cannot be fully accelerated using existing libraries. The result is that PyFibreBundle achieves performance suitable for use in real-time imaging systems.

B. Bundle Location, Cropping, and Masking

The first step of many image processing pipelines is to locate the fiber bundle within the image. The algorithm in PyFibreBundle for locating the bundle is fast, but it is designed only to work with a uniform or flat-field image (i.e., a reference image), and

not an image containing structure. If the image is color, the maximum value is first taken across all color channels to generate a monochrome image. A 2D Gaussian smoothing filter is then applied to remove the core pattern. The image is converted to 8 bit, and then the image is binarized using thresholding via Otsu's method. The largest connected region (other than the background) is then taken to be the bundle. The centroid of this region is assumed to be the center of the bundle and the radius as the smallest circle fitting within the region.

The image can then be cropped to show only the bundle by extracting a square image exactly enclosing the fitted circle. A mask can be created as a 2D array, which is 1 inside the circle and 0 outside the circle; multiplying subsequent images by this mask then sets all pixels outside the bundle to 0.

C. Determining the Core Spacing

It is useful to be able to detect the average spacing between cores in the bundle, as this can be used to automatically generate appropriate parameters for other processing methods that would otherwise need manual tuning. The approach taken in PyFibreBundle is to compute the spatial frequency domain representation of an image of the bundle and look for an apparent ring in the power spectrum that corresponds to the bundle core spacing.

Operationally, an average is taken across color channels (if present), the image is then cropped to the largest possible centered square, and a 2D Fourier transform is taken. The raw output of the Fourier transform is transformed so that the d.c. is in the center and highest spatial frequencies are at the edge of the image. The base-10 logarithm is then taken of the absolute value of the Fourier transform (i.e., log of the square root of the power spectrum), and a radial average is performed to generate a plot of amplitude against radial coordinate.

To find the ring corresponding to the spatial frequency of the core spacing, the radial profile is smoothed using a moving average filter, and then the discrete derivative is calculated. The first minimum of the resulting gradient is found, and then the first positive value of the gradient after this point is taken to be the spatial frequency of the core spacing. This can then be converted back to spatial units.

D. Locating the Fiber Cores

The fiber cores may need to be located for a number of purposes, including counting the number of cores in the bundle, and as one step of the calibration procedure for triangular linear interpolation (see Section 2.G).

The maximum value is taken across color channels (if present), and then a Gaussian smoothing filter is applied with a sigma equal to 20% of the estimated core spacing. This reduces noise without significantly weakening the core pattern. A morphological dilation is then performed using a circular structuring element with a diameter of 3 pixels. The effect of dilating a non-binary image is to leave local minima unchanged in value. Subtracting the dilated image from the original and inverting then leaves a bright spot at the center of each core.

The resulting image with core centers highlighted is then thresholded using Otsu's method, which cleanly separates the

core centers from the background. At this point, there may be multiple detected cores within each real core due to multimodal patterns causing multiple local maxima within a core. A morphological dilation with a circular structuring element with a diameter equal to one-third of the estimated core spacing merges these multiple detections within a core. All connected components are then found, with each component corresponding to a core. The centroid of each region then gives the location of the core.

E. Core Removal by Spatial Filtering

The simplest method of removing the core pattern is to apply a spatial filter to smooth the raw images. Three filtering methods are implemented in PyFibreBundle - Gaussian filter, median filter, and customized edge filter. The Gaussian and median filters are defined in the standard way. The edge filter is a smoothed step function that also allows a frequency domain low-pass filter to be defined. This is a radially symmetric, cosine-smoothed step function defined by two parameters: the spatial frequency of the step, R , and the smoothness (or skin thickness) of the step function, w , defined in terms of the spatial frequency difference between the function at 10% and 90% of maximum. Expressed in terms of the radial distance from the center of the spatial frequency domain plane, the filter is given by

$$f(r) = \begin{cases} 1, & \text{if } r < R \\ 0, & \text{if } r > R + w \\ \cos \left[\frac{\pi}{2w} (r - R) \right]^2, & \text{otherwise} \end{cases}$$

In practice, setting the R frequency to approximately twice the inverse of the core spacing yields good results.

F. Background Subtraction and Flat-Fielding

It is often beneficial to subtract a background image, for example in fluorescence endomicroscopy, to remove any fluorescence signal from the bundle itself. In PyFibreBundle, a background image can be provided that is then subtracted from the images to be processed. Almost all applications of fiber bundle imaging also benefit from flat-fielding, or normalization, to mitigate the impact of core-to-core variations in shape, size, and transmission. These are achieved simply by dividing by a reference image taken of a uniform target. If triangular linear interpolation is used, the background and flat-fielding correction are integrated into the interpolation procedure as described below.

G. Core Removal by Triangular Linear Interpolation

Triangular linear interpolation is a well-known approach to gridding irregularly space data. The implementation in PyFibreBundle is similar to those reported previously (e.g., [28]) and optimized to achieve very high frame rates (see Section 3.E).

A calibration is first performed using a flat-field image. In this calibration image, the bundle is located and masked as described in Section 2.B, and the locations of the cores are determined using the method described in Section 2.D. A Delaunay triangulation is then formed over the core positions. A reconstruction grid size is chosen, and for each pixel in the grid, the

enclosing triangle is found (or if the pixel is outside of the complex hull of the triangulation, this is recorded). The coordinates of the pixel are then converted to barycentric coordinates with respect to the vertices of the enclosing triangle. This concludes the calibration.

In the reconstruction stage, the intensity value of each core is extracted from the raw image. This can be done simply by taking the pixel value at the core location (which is rounded to the nearest integer). Optionally, a Gaussian smoothing filter may be applied first so that the pixel value at the core location contains a weighted average over the core.

For pixel j in the reconstructed image, we take the intensity values of the three cores in the surrounding triangle, c_1 , c_2 , and c_3 , and then we set the value of the pixel, p_j , using

$$p_j = \sum_{k=1}^3 c_k b_{j,k}, \quad (1)$$

where $b_{j,k}$ are the barycentric coordinates recorded for reconstruction grid pixel j in the calibration stage. Due to the use of barycentric coordinates, which are proportional to the inverse distances of the pixel from the cores at the triangle vertices, this results in triangular linear interpolation between the three cores.

If background and/or normalization images are being used, the value of each core in the background/normalization image is recorded at the calibration stage, using the same pre-smoothing filter. In the reconstruction stage, the extracted core values, c_i , are then corrected to c'_i prior to interpolation, using

$$c'_i = \frac{c_i - b_i}{n_i}, \quad (2)$$

where b_i and n_i are the core values from the background and normalization images, respectively.

H. Resolution Enhancement

Resolution enhancement is achieved by processing a stack of image in which the bundle is slightly shifted with respect to the sample. In the calibration stage, each image in the stack is first reconstructed using triangular linear interpolation to remove the core patterns. This step also determines the core locations, which are stored for later use. The relative shifts between the images are determined by computing the normalized cross-correlation between the first image and a template extracted from the center of each image. By default, the template is one-quarter the size of the image. The location of the peak of the normalized cross-correlation map is taken to be the shift between the images.

The core locations in each image are adjusted, using the measured shifts, to correct for the motion between images. The corrected core locations are then combined to form a much denser point cloud. A Delaunay triangulation is formed over this denser point cloud, and as before the location of each pixel in the reconstruction grid is recorded in Barycentric coordinates with respect to the three enclosing cores. This completes the calibration.

In the reconstruction stage, the intensity values of each core in each of the images is extracted, and the reconstruction grid pixel values are populated by triangular linear interpolation (as

described in Section 2.G), except that this is now performed using the denser set of points assembled from the shifted images. The result is a higher resolution image.

The calibration stage typically takes several seconds, while the reconstruction can be real-time on modest hardware (see Section 3.E for details). If the shifts between the images are repeatable, such as when a mechanical scanner is used (e.g., [39]), then the same calibration can be used for each set of shifted images, and so resolution enhancement can be obtained on live images.

I. Mosaicing

To stitch images together as the probe moves, the processed images are registered in a pairwise fashion using normalized cross-correlation. The images can optionally be downsized first for speed. By default, a square with a side length equal to half the bundle diameter is extracted from the first image, and a square with side lengths equal to a quarter of the bundle diameter is extracted from the second—this is the template. Normalized cross-correlation is then performed between the template and cropped image, and the peak of the correlation map is identified. The offset of this peak from the center of the cross-correlation map provides the shift.

Each image is then added to the mosaic with the detected shift from the previous image applied. By default, images are blended into the existing mosaic using a cosine window, but they can also be added dead-leaf for increased speed (i.e., without any blending).

Since the mosaic image has finite size, eventually the inserted image may reach the edge of the mosaic image. In this case, there are several options within PyFibreBundle. This can simply be ignored, allowing the mosaic to be cropped. The mosaic image can be dynamically expanded to accommodate the image location, or the mosaic can be scrolled, losing any information on the opposite edge.

If the images have a very small shift, as often happens if the fiber bundle is close to being non-moving, continuously blending images leads to an unwanted blurring effect. To avoid this, PyFibreBundle can be set to only add an image when there is a minimum shift, which by default is 25 pixels.

At times the probe may move too fast to be registered, and the best detected shift will not correspond to the real shift. In these cases, it may be desirable that the mosaic is reset and begins again. PyFibreBundle allows the value of the peak of the cross-correlation to be monitored and the mosaic to be reset if this drops below a threshold. The intensity and sharpness of the images can also be monitored so that if the quality drops below a certain threshold (e.g., because the fiber bundle probe has lost contact with the sample being imaged) the mosaic is also reset.

J. Imaging Systems

To evaluate PyFibreBundle, three fiber bundle imaging systems were used: a transmission monochrome endomicroscope, a color endomicroscope, and a line-scan confocal fluorescence. All three systems used a Fujikura 30,000 core fiber bundle (FIGH-30-650S), with a 600 μm active imaging area. No lens

was used at the distal end of the bundle, and so the distal tip was placed in direct contact with the sample.

In the transmission monochrome system, the proximal end of the fiber bundle was imaged directly onto a CMOS camera (FLIR Flea3 FL3-U3-13S2M-CS) by a 10X infinity corrected microscope objective and a 100 mm focal length tube lens. The magnification factor between the fiber bundle and the camera was approximately 6, and the camera pixel size was 3.63 μm ; the bundle active area was, therefore, approximately 900 pixels in diameter on the camera. For transmission imaging, samples were back-illuminated by a Royal Blue LED.

The color endomicroscope is similar in concept to the fiber bundle endocytoscope concept previously published [48], except that it was built using a Raspberry Pi v4 computer and Raspberry Pi camera module v2. This camera module employs the Sony IMX219 chip with a pixel size of 1.12 μm arranged in a 3280 \times 2464 grid. The images were down-sampled to 1024 \times 800 using the camera firmware. A 10X objective and a 50 mm focal length tube lens was used to achieve a magnification factor of approximately 3, and the bundle therefore covered approximately 750 pixels on the camera after down-sampling. For the images used here, the sample was simply directly back-illuminated using a white LED rather than via a fiber as in [48].

The line-scanning confocal fluorescence system is based on a design first reported in [10] and built on in [49]. Unlike a fully confocal endomicroscope, the system does not scan a focused spot over the sample in 2D; instead, a line generated from a 488 nm laser is scanned in 1D over the bundle using a galvo scanning mirror. This scanning line is transmitted to the sample via the bundle. Fluorescence from the sample, which is excited by the 488 nm laser, returns along the bundle, is band-pass filtered to remove any reflected 488 nm light, and is then imaged onto a rolling shutter CMOS camera. The rolling shutter of the camera is synchronized with the scanning laser line so that the rolling shutter acts as a moving slit for confocal sectioning. The system uses the same components as the design reported in [49], except that the components for generating two images of the bundle on the camera were removed and replaced with a 100 mm focal length tube lens. The system, therefore, has the same magnification as the transmission monochrome setup.

3. RESULTS AND DISCUSSION

All raw data and scripts used to generate the results presented below are permanently archived in [50]. This archive also contains a copy of PyFibreBundle Release 1.3.4.

A. Bundle Location, Cropping, and Masking

Figure 1(a) shows a raw image from the monochrome transmission endomicroscopy system with no object in the field of view, giving essentially an undisturbed flat-field image. Individual fiber cores can easily be seen. The average core spacing was determined by a PyFibreBundle function to be 3.2 μm . A manual measurement was also made using ImageJ, averaging 10 measurements of core separations, to give $(3.5 \pm 0.1) \mu\text{m}$.

PyFibreBundle was used to detect the bundle location and size in the image, and the result is shown by the dashed circle

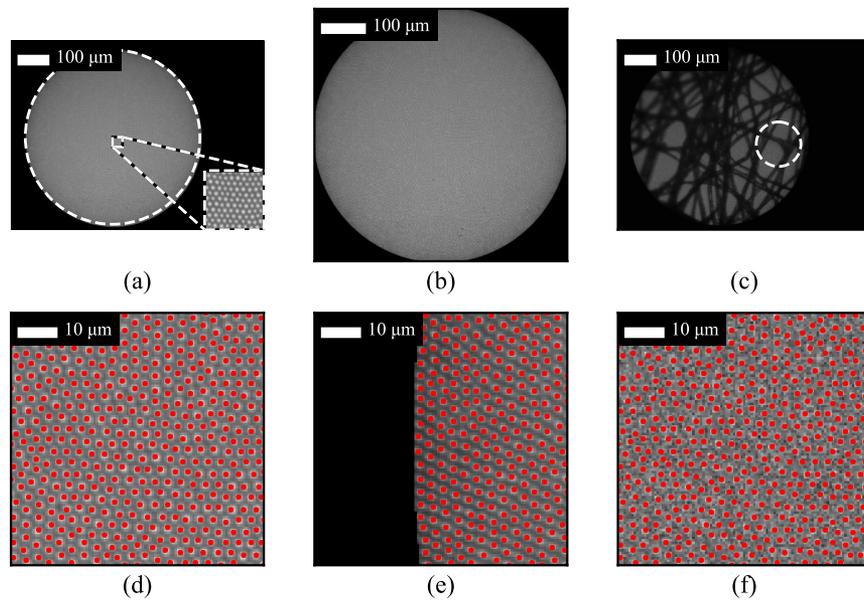


Fig. 1. Demonstration of bundle and core locating routines. (a) Raw flat-field image with zoom showing core pattern. The dashed circle around the bundle is the automatically determined bundle location. (b) Raw image automatically cropped to bundle. (c) Bundle detection applied to image with structure, showing that the algorithm fails. (d) Automatically determined core locations, showing a zoom on an area of approximately $68 \times 68 \mu\text{m}$ at the center of the bundle. (e) As (d) but showing the left hand edge. (f) Core locations determined from noisier image, same area as (d).

in Fig. 1(a). PyFibreBundle determined a radius of 445 pixels, compared to a manual measurement of 450 pixels made in ImageJ. The manual measurement included the entirety of the cores on the outer edge, whereas PyFibreBundle finds an area closer to the separation between the center of the extreme cores, explaining this discrepancy. PyFibreBundle was then used to crop the image to show only the bundle and to mask areas outside the bundle, with the result shown in Fig. 1(b). As discussed in Section 2.B, the bundle detection routine is only designed to work on flat-field calibration images and not on images with structure; Fig. 1(c) shows that the detection fails completely on an image of tissue paper.

The location of each core within the bundle can also be located automatically, and this is essential for the linear interpolation core removal method. The routine for core finding was applied to the image from Fig. 1(a), and two subsets of the results are shown in Fig. 1(d) (the center of the bundle) and Fig. 1(e) (the edge of the bundle). A total of 27,258 cores were found, and visual inspection suggests that there are few erroneous detections or missed cores. Since the ground truth location of the cores is not known, the accuracy cannot be quantified, but for the case of Fig. 1(d), all the core locations were manually confirmed to lie within 1 pixel of the apparent center of the core. The routine requires an estimate of the core spacing; the core spacing value determined by PyFibreBundle was used, keeping the process fully automated. Note that the bundle was automatically masked to set pixels outside of the bundle to 0 prior to detecting the cores. This results in a small number of cores outside of the detected bundle diameter not being detected. The core location is somewhat robust to noise, and Fig. 1(f) shows the algorithm applied to the same image with added Gaussian noise with a standard deviation equal to 20% of the mean intensity inside the bundle. Even though the image appears noisy and the cores are

much harder to discern by eye, they are still identified well by the algorithm, with a total of 26,574 cores found.

B. Core Pattern Removal

The same tissue paper image as used in Fig. 1 was first automatically cropped and masked using the calibration image and normalized using the calibration image, and then a Gaussian filter of different sizes was applied. As can be seen in Fig. 2(a), even without a filter, the honeycomb pattern becomes less visible simply through normalization. As the filter size (Gaussian σ value) increases, the core pattern is better removed at the expense of eventually blurring the image. A filter size of $1.5 \mu\text{m}$ is sufficient to completely remove the pattern; this is approximately half the core-core spacing.

Figure 3 shows the results of using the linear interpolation method on the same raw image, both with and without normalization (flat-fielding). Each column shows the effect of a different size Gaussian pre-filter prior to extracting the core-intensities, i.e., changing the degree to which the outer parts of each core were included in the calculation. It can be seen that this makes little difference in practice, at least for this particular image, up until a filter size of $1.5 \mu\text{m}$, at which point there is visible blurring in the inset. The application of the method to color images is shown in Fig. 4. In this case, an option within PyFibreBundle to individually normalize each color channel was used, and this has the effect of performing a simple white-balancing to remove spatially dependent chromatic effects, as can be clearly seen in the results.

C. Mosaicing

An example mosaic from confocal linescan fluorescence endomicroscopy of stained tissue paper is shown in Fig. 5. The

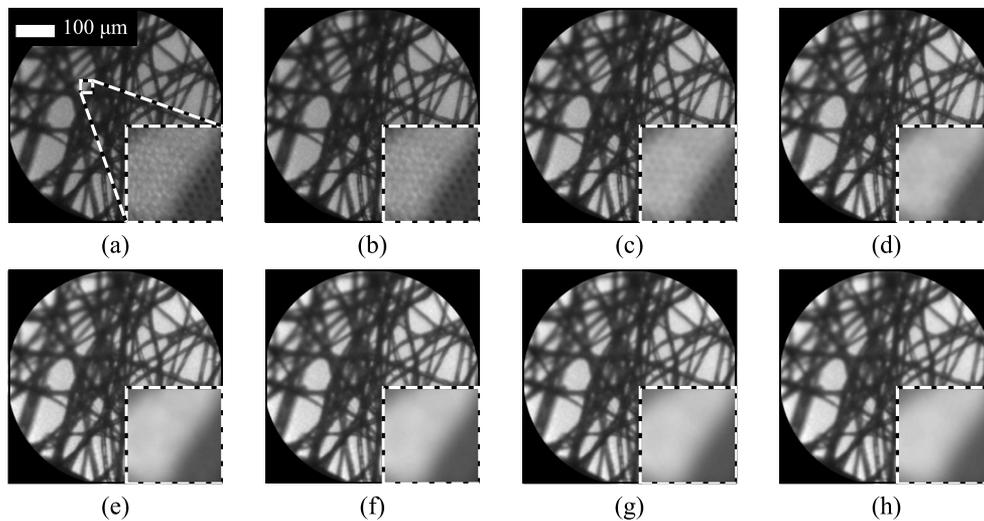


Fig. 2. Core removal using Gaussian filters of different sizes for transmission image of lens tissue paper. Raw images were first cropped, masked and normalized. (a) has no further processing, while (b)–(h) have a Gaussian filter σ values of (b) 0.5 μm , (c) 1.0 μm , (d) 1.5 μm , (e) 2.0 μm , (f) 2.5 μm , (g) 3.0 μm , and (h) 3.5 μm . The equivalent size in pixels was (b) 0.74, (c) 1.5, (d) 2.2, (e) 3.0, (f) 3.7, (g) 4.5, and (h) 5.2 pixels.

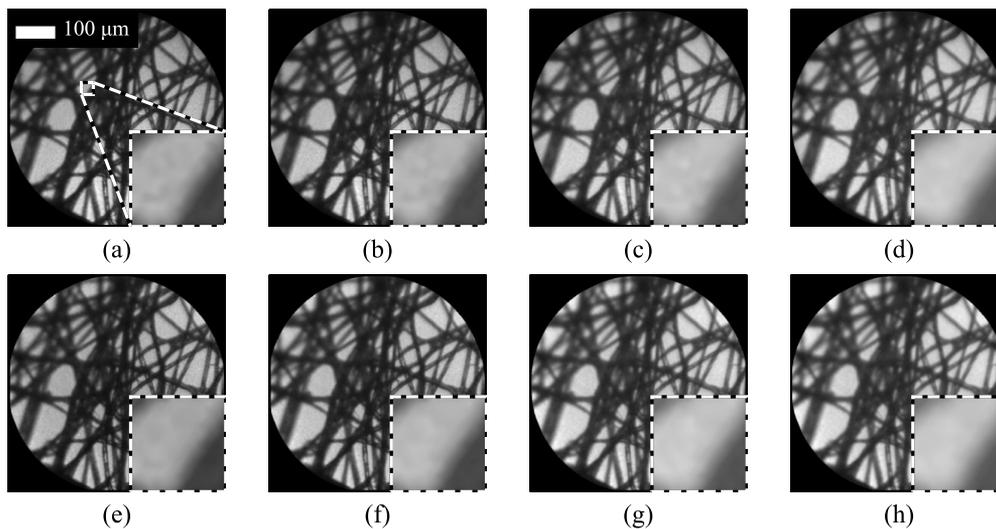


Fig. 3. Core removal using triangular linear interpolation for transmission image of lens tissue paper. (a)–(d) The top row was reconstructed without normalization, and (e)–(h) the bottom row was reconstructed with normalization by the calibration image. The effect of different smoothing filters prior to core value extraction is shown across the columns.

fiber bundle was manually drawn across the tissue paper while images were captured and saved at 120 fps. Core-removal, background subtraction, and normalization was performed using the linear interpolation method with no pre-filter. Figure 5(a) shows a single image for reference, while Fig. 5(b) shows an example of a short mosaic created using the dead-leaf approach (i.e., no blending of frames). The zoom inset shows that the boundary between two images can be observed. This effect is removed by blending of the frames, as shown in Fig. 5(c). Down-sampling from 512×512 to 200×200 prior to mosaicing increased speed, but results in a slight loss of detail, as can be seen in Fig. 5(d). The mosaicing results indicated that the probe was, on average, moving at 0.04 mm per frame, or 4.8 mm/s, with a peak speed of 15 mm/s. In practice, the algorithm can only cope with inter-frame shifts of up to approximately 1/4 the image diameter

to enable sufficient overlap for image cross-correlation, corresponding to a maximum speed of 18 mm/s for this particular system.

D. Resolution Enhancement

A set of 8 images was acquired using the monochrome transmission system, with the bundle manually shifted with respect to a USAF resolution target with elements down to Group 9 Element 3. Shifts were made randomly, within a maximum shift of approximately 30 μm . The resolution enhancement technique was then applied using 2, 4, and all 8 images. The resulting images are compared with the reconstruction from a single image in Fig. 6. The zoomed insets, which show Groups 8 and 9, demonstrate an obvious improvement in resolution and line pair contrast, with visibility improving from approximately Group 7 Element 3 to Group 8 Element 1 (161 lp/mm

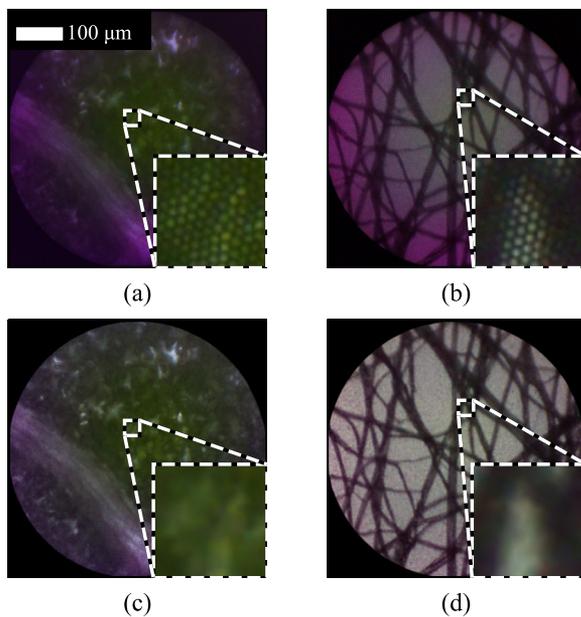


Fig. 4. Core removal from color images using linear interpolation. (a) and (b) are cropped raw images from a leaf and lens tissue paper, respectively, and (c) and (d) are the corresponding processed images. Each color channel was individually normalized, improving the color balance compared with the raw images.

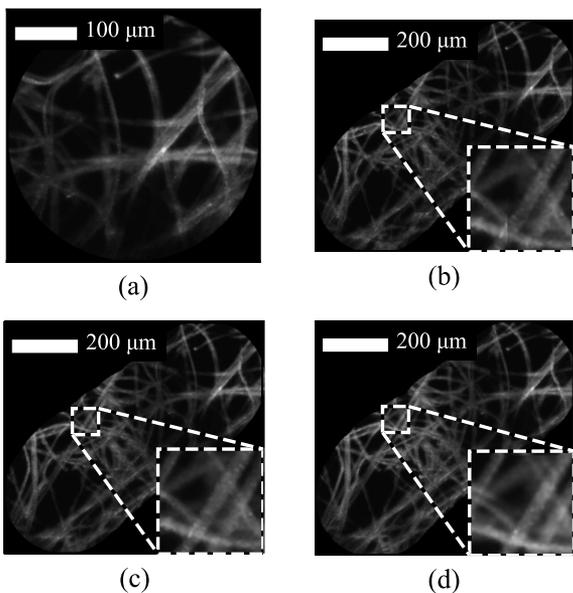


Fig. 5. Example mosaicing of confocal fluorescence images of stained lens tissue paper. (a) Single image frame. (b) Mosaicing using dead-leaf approach; the image join is visible in the inset. (c) Mosaicing using blending; the image join is no longer visible in the inset. (d) Mosaicing using blending and down-sampling for higher speed; a small loss of detail can be seen in the inset.

to 256 lp/mm) when all 8 images are used, and the contrast of Group 6 Element 1 (the smaller of the differences between central black and the two neighboring white lines, normalized to average intensity across the profile) improving from 0.0 to 0.23. Much of the visual improvement is also obtained when using 4 images (although contrast only improves to 0.08), while even

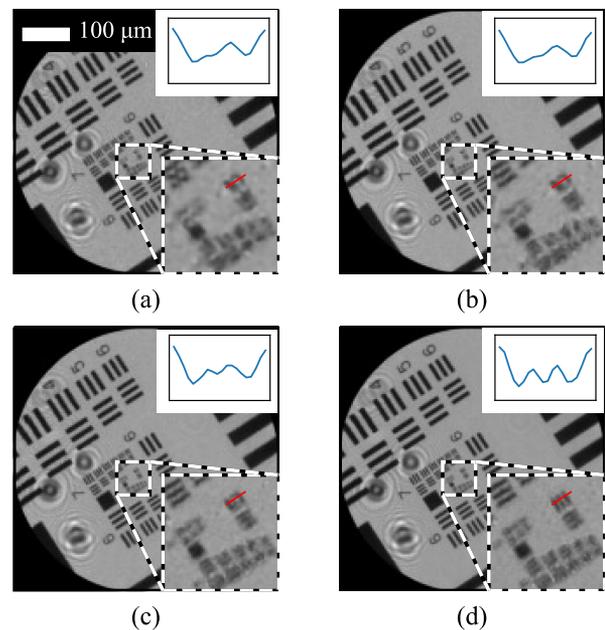


Fig. 6. Resolution enhancement applied to monochrome transmission image of USAF resolution target. The target was manually shifted with respect to the bundle, and 8 images were acquired. (a) Reconstruction of a single image using triangular linear interpolation. (b)–(d) Reconstruction with resolution enhancement using (b) 2, (c) 4, and (d) all 8 of the images. The upper right inset on each image is a profile taken across Group 6 Element 1, as indicated by the line on the images.

using 2 images provides some visual improvement for the larger line pairs.

E. Benchmarking

A selection of the core functions of PyFibreBundle was tested on three machines: a desktop PC (Intel Core i7-7700, 4 cores, 24 GB RAM) running Windows 10, a laptop (Intel Core i5-8250U, 4 cores, 8 GB RAM) running Windows 10, and a Raspberry Pi 4 Model 4 (Cortex-A72, 4 cores, 4 GB RAM) running Raspberry Pi OS 64 bit (based on Debian Linux). The PC was installed with a GPU, but PyFibreBundle does not make use of this. All tests used 8 bit raw images and do not include time to display results. Timings are averages across at least 7 runs after allowing for warm-up, initialization, and any required runs of the Numba JIT Compiler.

The results are summarized in Table 1. On the PC and laptop computers, frame rates of over 100 Hz are readily achievable for various types of processing. In particular, using a grid of 512×512 for linear interpolation allows a frame rate of approximately 500 Hz on the PC, which is faster than that required for any endomicroscopy system reported to-date. Mosaicing can also be at similar speeds if images are first down-sampled to 200×200 images. Resolution enhancement using 8 images onto a 800×800 grid is readily achievable at video rates on both the PC and laptop. When tested within a lightweight PyQt-based GUI developed in-house, real-time processing using linear interpolation was still feasible at 120 fps.

While a Raspberry Pi has previously been used to capture and perform inference on endomicroscopy images [51], the capability to perform core-pattern removal by interpolation

Table 1. Results of Speed Tests on Three Computers, with Specifications as Described in the Paper^a

		Time in ms		
		PC	Laptop	Raspberry Pi
Linear interpolation	512 × 512 Calibration	380	450	1200
	512 × 512 Reconstruction	1.9	2.6	8.6
	1024 × 1024 Calibration	580	720	1700
Gaussian filter	1024 × 1024 Reconstruction	8.4	12	34
	Filter Only	5.8	9.3	13
Resolution enhancement	Filter + Normalization	13	20	29
	Calibration	15,000	18,000	19,000
Mosaicing	Reconstruction	11	17	49
	512 × 512 Images	17	21	66
	200 × 200 Images	1.0	2.0	14

^aAll times given to 2 significant figures. Numbers such as 512 × 512 refer to image sizes in pixels. The Gaussian filter had a sigma of 2 pixels and was applied to a 920 × 920 pixel image. The resolution enhancement used 8 shifted images and was reconstructed on a grid of 800 × 800 pixels. Mosaicing was performed onto a 2000 × 2000 pixel mosaic image, with blending.

at video rates has not previously been demonstrated on such a low-end computer. On the Raspberry Pi, PyFibreBundle achieved an equivalent frame rate of 100 fps for linear interpolation onto a 512 × 512 grid. In a complete system, there would be some additional overhead for image acquisition and display, but the video-rate display is nevertheless feasible. Mosaicing with full resolution on the Raspberry Pi is slightly slow for live use at 66 ms per frame. However, with down-sampling to 200 × 200 pixels, total time for both interpolation and mosaicing was approximately 23 ms on the Raspberry Pi, suggesting that real-time video mosaicing would also be possible with a small loss of resolution.

4. CONCLUSION

This paper demonstrates that the PyFibreBundle Python package can process raw images from monochrome and color endomicroscopes, removing the core pattern as well as subtracting background signal and flat-fielding to correct for core-to-core variations in light transmission. Mosaicing can be employed to increase the effective field-of-view, and resolution can be improved by combining multiple shifted images. The package is fast enough to be used for live imaging on consumer-grade PCs and even on a Raspberry Pi single board computer, and it therefore may be a useful resource for anyone developing open and low-cost fiber bundle imaging systems.

The package currently only implements a small subset of the many published approaches to fiber bundle image processing and analysis, as reviewed in [25]. In particular, it implements several methods for removing the core patterns, with a focus on those that are suitable for real-time use. While triangular linear interpolation holds several advantages, including compatibility with resolution enhancement, it requires a calibration process and, further, that there are no small shifts of the proximal end of the bundle with respect to the camera. There may, therefore, be some applications for which the simpler spatial filtering methods would be preferred.

There is considerable scope for further development of the package. While alternative approaches to core pattern removal are mostly either unsuitable for real-time use or offer little further benefit in comparison with linear interpolation, a great variety of other techniques could offer value to the scientific

community if integrated with PyFibreBundle. For example, it has been shown that exploiting spatial information within the cores can lead to a primitive form of light-field imaging [19], and much improved mosaicing performance is possible with algorithms that allow for non-rigid deformations of the tissue. The package is open-source and fully documented; hence, other research teams can easily contribute their code in order to help grow this resource further for the benefit of the biomedical imaging community.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [50].

REFERENCES

1. C. B. Morgenthal, W. O. Richards, B. J. Dunkin, K. A. Forde, G. Vitale, and E. Lin, and for the SAGES Flexible Endoscopy Committee, "The role of the surgeon in the evolution of flexible endoscopy," *Surg. Endosc.* **21**, 838–853 (2007).
2. D. L. Dickensheets and G. S. Kino, "Micromachined scanning confocal optical microscope," *Opt. Lett.* **21**, 764–766 (1996).
3. S. Lemire-Renaud, M. Rivard, M. Strupler, D. Morneau, F. Verpillat, X. Daxhelet, N. Godbout, and C. Boudoux, "Double-clad fiber coupler for endoscopy," *Opt. Express* **18**, 9755–9764 (2010).
4. A. F. Gmitro and D. Aziz, "Confocal microscopy through a fiber-optic imaging bundle," *Opt. Lett.* **18**, 565–567 (1993).
5. B. Viellerobe, A. Osdoit, C. Cavé, F. Lacombe, S. Loiseau, and B. Abrat, "Mauna Kea technologies' F400 prototype: a new tool for *in vivo* microscopic imaging during endoscopy," *Proc. SPIE* **6082**, 39–49 (2006).
6. A. Fugazza, F. Gaiani, M. C. Carra, F. Brunetti, M. Lévy, I. Sobhani, D. Azoulay, F. Catena, G. L. de'Angelis, and N. de'Angelis, "Confocal laser endomicroscopy in gastrointestinal and pancreatobiliary diseases: a systematic review and meta-analysis," *BioMed Res. Int.* **2016**, e4638683 (2016).
7. M. C. Pierce, P. M. Vila, A. D. Polydorides, R. Richards-Kortum, and S. Anandasabapathy, "Low-cost endomicroscopy in the esophagus and colon," *Am. J. Gastroenterol.* **106**, 1722–1724 (2011).
8. T. J. Muldoon, D. Roblyer, M. D. Williams, V. M. Stepanek, R. Richards-Kortum, and A. M. Gillenwater, "Noninvasive imaging of oral neoplasia with a high-resolution fiber-optic microendoscope," *Head Neck* **34**, 305–312 (2012).
9. Y. S. Sabharwal, A. R. Rouse, L. Donaldson, M. F. Hopkins, and A. F. Gmitro, "Slit-scanning confocal microendoscope for high-resolution *in vivo* imaging," *Appl. Opt.* **38**, 7133–7144 (1999).

10. M. Hughes and G.-Z. Yang, "Line-scanning fiber bundle endomicroscopy with a virtual detector slit," *Biomed. Opt. Express* **7**, 2257–2268 (2016).
11. N. Bozinovic, C. Ventalon, T. Ford, and J. Mertz, "Fluorescence endomicroscopy with structured illumination," *Opt. Express* **16**, 8016–8025 (2008).
12. P. A. Keahey, T. S. Tkaczyk, K. M. Schmeler, and R. R. Richards-Kortum, "Optimizing modulation frequency for structured illumination in a fiber-optic microendoscope to image nuclear morphometry in columnar epithelium," *Biomed. Opt. Express* **6**, 870–880 (2015).
13. A. D. Thrapp and M. R. Hughes, "Automatic motion compensation for structured illumination endomicroscopy using a flexible fiber bundle," *J. Biomed. Opt.* **25**, 026501 (2020).
14. H. Makhlof, A. F. Gmitro, A. A. Tanbakuchi, J. A. Udovich, and A. R. Rouse, "Multispectral confocal microendoscope for *in vivo* and *in situ* imaging," *J. Biomed. Opt.* **13**, 044016 (2008).
15. W. Göbel, J. N. D. Kerr, A. Nimmerjahn, and F. Helmchen, "Miniaturized two-photon microscope based on a flexible coherent fiber bundle and a gradient-index lens objective," *Opt. Lett.* **29**, 2521–2523 (2004).
16. S. Cheng, J. J. Rico-Jimenez, J. Jabbour, B. Malik, K. C. Maitland, J. Wright, Y.-S. L. Cheng, and J. A. Jo, "Flexible endoscope for continuous *in vivo* multispectral fluorescence lifetime imaging," *Opt. Lett.* **38**, 1515–1517 (2013).
17. L. Wurster, A. Kumar, D. Fechtig, L. Ginner, and R. Leitgeb, "Lensless holographic endoscopy with a fiber bundle," in *Optical Tomography and Spectroscopy* (Optical Society of America, 2016), paper OTu4C–5.
18. M. R. Hughes, "Inline holographic microscopy through fiber imaging bundles," *Appl. Opt.* **60**, A1–A7 (2021).
19. A. Orth, M. Ploschner, E. Wilson, I. Maksymov, and B. Gibson, "Optical fiber bundles: ultra-slim light field imaging probes," *Sci. Adv.* **5**, eaav1555 (2019).
20. L. M. Wurster, L. Ginner, A. Kumar, M. Salas, A. Wartak, and R. A. Leitgeb, "Endoscopic optical coherence tomography with a flexible fiber bundle," *J. Biomed. Opt.* **23**, 066001 (2018).
21. R. Juškattis, T. Wilson, and T. F. Watson, "Real-time white light reflection confocal microscopy using a fibre-optic bundle," *Scanning* **19**, 15–19 (1997).
22. M. R. Hughes, P. Giataganas, and G.-Z. Yang, "Color reflectance fiber bundle endomicroscopy without back-reflections," *J. Biomed. Opt.* **19**, 030501 (2014).
23. T. N. Ford, K. K. Chu, and J. Mertz, "Phase-gradient microscopy in thick tissue with oblique back-illumination," *Nat. Methods* **9**, 1195–1197 (2012).
24. E. R. Andresen, S. Sivankutty, V. Tsvirkun, G. Bouwmans, and H. Rigneault, "Ultrathin endoscopes based on multicore fibers and adaptive optics: a status review and perspectives," *J. Biomed. Opt.* **21**, 121506 (2016).
25. A. Perperidis, K. Dhaliwal, S. McLaughlin, and T. Vercauteren, "Image computing for fibre-bundle endomicroscopy: a review," *Medical Image Analysis* **62**, 101620 (2020).
26. M. J. Suter, J. M. Reinhardt, P. R. Montague, P. Taft, J. Lee, J. Zabner, and G. McLennan, "Bronchoscopic imaging of pulmonary mucosal vasculature responses to inflammatory mediators," *J. Biomed. Opt.* **10**, 034013 (2005).
27. C. Winter, S. Rupp, M. Elter, C. Munzenmayer, H. Gerhauser, and T. Wittenberg, "Automatic adaptive enhancement for images obtained with fiberoptic endoscopes," *IEEE Trans. Biomed. Eng.* **53**, 2035–2046 (2006).
28. G. Le Goualher, A. Perchant, M. Genet, C. Cavé, B. Viellerobe, F. Berier, B. Abrat, and N. Ayache, *Towards Optical Biopsies with an Integrated Fibered Confocal Fluorescence Microscope* (Springer, 2004), pp. 761–768.
29. T. Vercauteren, F. Doussoux, M. Cazaux, G. Schmid, N. Linard, M.-A. Durin, H. Gharbi, and F. Lacombe, "Multicolor probe-based confocal laser endomicroscopy: a new world for *in vivo* and real-time cellular imaging," *Proc. SPIE* **8575**, 857504 (2013).
30. X. Liu, L. Zhang, M. Kirby, R. Becker, S. Qi, and F. Zhao, "Iterative l_1 -min algorithm for fixed pattern noise removal in fiber-bundle-based endoscopic imaging," *J. Opt. Soc. Am. A* **33**, 630–636 (2016).
31. J.-H. Han, S. M. Yoon, and G.-J. Yoon, "Decoupling structural artifacts in fiber optic imaging by applying compressive sensing," *Optik* **126**, 2013–2017 (2015).
32. T. Vercauteren, A. Meining, F. Lacombe, and A. Perchant, "Real time autonomous video image registration for endomicroscopy: fighting the compromises," *Proc. SPIE* **6861**, 90–97 (2008).
33. N. Bedard, T. Quang, K. Schmeler, R. Richards-Kortum, and T. S. Tkaczyk, "Real-time video mosaicing with a high-resolution microendoscope," *Biomed. Opt. Express* **3**, 2428–2435 (2012).
34. T. Vercauteren, A. Perchant, G. Malandain, X. Pennec, and N. Ayache, "Robust mosaicing with correction of motion distortions and tissue deformations for *in vivo* fibered microscopy," *Med. Image Anal.* **10**, 673–692 (2006).
35. M. Hughes and G.-Z. Yang, "High speed, line-scanning, fiber bundle fluorescence confocal endomicroscopy for improved mosaicking," *Biomed. Opt. Exp.* **6**, 1241–1252 (2015).
36. P. Giataganas, M. Hughes, C. J. Payne, P. Wisanuvej, B. Temelkuran, and G.-Z. Yang, "Intraoperative robotic-assisted large-area high-speed microscopic imaging and intervention," *IEEE Trans. Biomed. Eng.* **66**, 208–216 (2019).
37. F. Berier and A. Perchant, "Method and system for super-resolution of confocal images acquired through an image guide, and device used for implementing such a method," U.S. patent 7,646,938 (12 January 2010).
38. M. Kyriash, R. Kester, R. Richards-Kortum, and T. Tkaczyk, "Improving spatial resolution of a fiber bundle optical biopsy system," *Proc. SPIE* **7558**, 755807 (2010).
39. K. Vyas, M. Hughes, B. G. Rosa, and G.-Z. Yang, "Fiber bundle shifting endomicroscopy for high-resolution imaging," *Biomed. Opt. Express* **9**, 4649–4664 (2018).
40. Y. Huang, W. Zhou, B. Xu, J. Liu, D. Xiong, and X. Yang, "Resolution improvement in real-time and video mosaicing for fiber bundle imaging," *OSA Contin.* **4**, 2577–2590 (2021).
41. H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graphical Models Image Process.* **54**, 181–186 (1992).
42. C. Renteria, J. Suárez, A. Licudine, and S. A. Boppart, "Depixelation and enhancement of fiber bundle images by bundle rotation," *Appl. Opt.* **59**, 536–544 (2020).
43. M. Eadie, J. Liao, W. Ageeli, G. Nabi, and N. Krstajić, "Fiber bundle image reconstruction using convolutional neural networks and bundle rotation in endomicroscopy," *Sensors* **23**, 2469 (2023).
44. D. Ravi, A. B. Szczotka, D. I. Shafir, S. P. Pereira, and T. Vercauteren, "Effective deep learning training for single-image super-resolution in endomicroscopy exploiting video-registration-based reconstruction," *Int. J. Comput. Assist. Radiol. Surg.* **13**, 917–924 (2018).
45. J. Shao, J. Zhang, R. Liang, and K. Barnard, "Fiber bundle imaging resolution enhancement using deep learning," *Opt. Express* **27**, 15880–15890 (2019).
46. D. Norberg, "Source code and example data for 'Open source image processing methods for real-time fibre bundle optical endomicroscopy,'" Edinburgh DataShare 2022, <https://datashare.ed.ac.uk/handle/10283/3803>.
47. M. R. Hughes, "Fibre bundle image processing/core removal (Matlab)," Mathworks, 2023, <https://uk.mathworks.com/matlabcentral/fileexchange/75248-fibre-bundle-image-processing-core-removal-matlab>.
48. M. Hughes, T. P. Chang, and G.-Z. Yang, "Fiber bundle endocytoscopy," *Biomed. Opt. Express* **4**, 2781–2794 (2013).
49. A. D. Thrapp and M. R. Hughes, "Reduced motion artifacts and speed improvements in enhanced line-scanning fiber bundle endomicroscopy," *J. Biomed. Opt.* **26**, 056501 (2021).
50. M. Hughes, "Real-timing processing of fiber bundle endomicroscopy images in Python using pyfibrebundle: data and code," figshare, 2023, <https://dx.doi.org/10.6084/m9.figshare.23932872>.
51. S. Parra, E. Carranza, J. Coole, B. Hunt, C. Smith, P. Keahey, M. Maza, K. Schmeler, and R. Richards-Kortum, "Development of low-cost point-of-care technologies for cervical cancer prevention based on a single-board computer," *IEEE J. Transl. Eng. Health Med.* **8**, 1–10 (2020).