[32] Digital Autofocus Methods for Automated Microscopy

By Feimo Shen, Louis Hodgson, and Klaus Hahn

Abstract

Automatic focusing of microscope images is an essential part of modern high-throughput microscopy. This chapter describes implementation of a robust autofocus system appropriate for using either air or oil immersion objectives in robotic imaging. Both hardware and software algorithms are described, and caveats of using viscous immersion media with multifield scanning are detailed.

Introduction

In traditional microscopy, it can be difficult to maintain focus when using objectives with high numerical aperture (NA) and low working distances. This problem is compounded during extended time-lapse studies when temperature fluctuations affect focus. Focus is especially critical in time-lapse high-content screening because this requires scanning of many fields of view (FOV) over large areas (Bajaj *et al.*, 2000; Shen *et al.*, 2006). High NA objectives characteristically have a very low depth of field (DOF); only a very thin section of the specimen falls in focus in the acquired image. In order to solve the problem of focal drift, a fast and robust autofocus method is required.

Potential sources of focus fluctuations are (1) the substrate supporting the specimen is not flat; (2) thermal fluctuation causes the distance between the sample and the optical elements to drift over time; and (3) fluid flow and exchange of medium in cell perfusion experiments cause vibration. Specimen supports are typically not flat. For example, the height of a 9 × 15-mm^2 glass coverslip can vary 6 μm across the diagonal (Bravo-Zanoguera *et al.*, 1998). The specimen support used in larger area scans can vary even more, especially when level mounting of the specimen is difficult or there is a misalignment of the scanning stage. For microscopes that are aligned carefully, thermal fluctuations are the major culprit. The focal drift associated with temperature fluctuation can be as much as 1 μm per degree Celsius (data not shown). In live-cell perfusion experiments, microfluctuations in temperature additionally affect the focus position. Autofocus is essential in time-lapse studies to overcome these problems.

METHODS IN ENZYMOLOGY, VOL. 414 0076-6879/06 $35.00

DOI: 10.1016/S0076-6879(06)14032-X

Commercially available options for autofocus can be categorized into three groups: (1) hardware-based solutions, (2) software-based solutions, and (3) fully integrated high-throughput microscopy (HTM) platforms. Hardware-based solutions that use RS-170 video signals include the MAC series of video autofocus processors from Ludl Electronic Products Ltd. (Hawthorne, NY) and ASI's video autofocus (Applied Scientific Instrumentation, Inc., Eugene, OR). Most makes of microscopy integration software include software-based autofocus in their packages. Example systems that allow a certain level of user customization include MetaMorph 6.3 (Molecular Devices Corp, Sunnyvale, CA) and SlideBook 4.1 (Intelligent Imaging Innovations, Inc., Denver, CO). Integrated HTM platforms such as the KineticScan HCS Reader (Cellomics, Inc., Pittsburgh, PA), Pathway (BD Biosciences, Rockwille, MD), and CellLab IC 100 (Beckman Coulter, Inc., Fullerton, CA) are easy-to-use, black-box systems that have autofocus capability.

All these premanufactured solutions are not readily adapted to many individual user requirements. Perhaps most importantly, they are ill suited to high-resolution imaging of dim fluorescence because they do not provide for using higher NA oil immersion lenses. Most can accommodate only dry objective lenses up to 40×, 0.95 NA. This chapter details the construction of an autofocus system that overcomes this difficulty using off-the-shelf materials and methods that can be engineered into any microscope for automation.

In the system described here, the hardware acquires a *z* stack of images for each focus attempt. Digital image processing software determines the high-frequency content of each image in the stack and uses the results to determine a "focus index" for each image. The sharpest image is acquired at the best focus position calculated from the focus indices.

Hardware

The components needed to carry out digital autofocus in a microscopy scan are as follows:

1. A microscope capable of differential interference contrast (DIC) imaging.
2. Hardware to precisely control the spacing between the objective and the specimen.
3. A motorized *x–y* stage with linear encoding feedback accuracy.
4. An equivalent of a computer with a Pentium 4 2.4-GHz central processing unit (CPU), 2-GB random access memory (RAM), and minimum of two peripheral component interconnect (PCI) expansion slots.
5. A scientific grade digital camera that can interface with a computer.

Figure 1 shows an example of an assembled system in our laboratory. We have an Olympus Model IX71 inverted microscope (Olympus America, Inc., Melville, NY) on which we have fitted the autofocus module. This is an infinity-corrected microscope equipped to perform DIC imaging. DIC imaging is preferred over other means of generating contrast because it does not attenuate fluorescence, is easy to set up, and has greater high-frequency content (Inoue and Spring, 1997). The latter is very important in our resolution-based autofocus. In contrast to DIC, bright-field images do

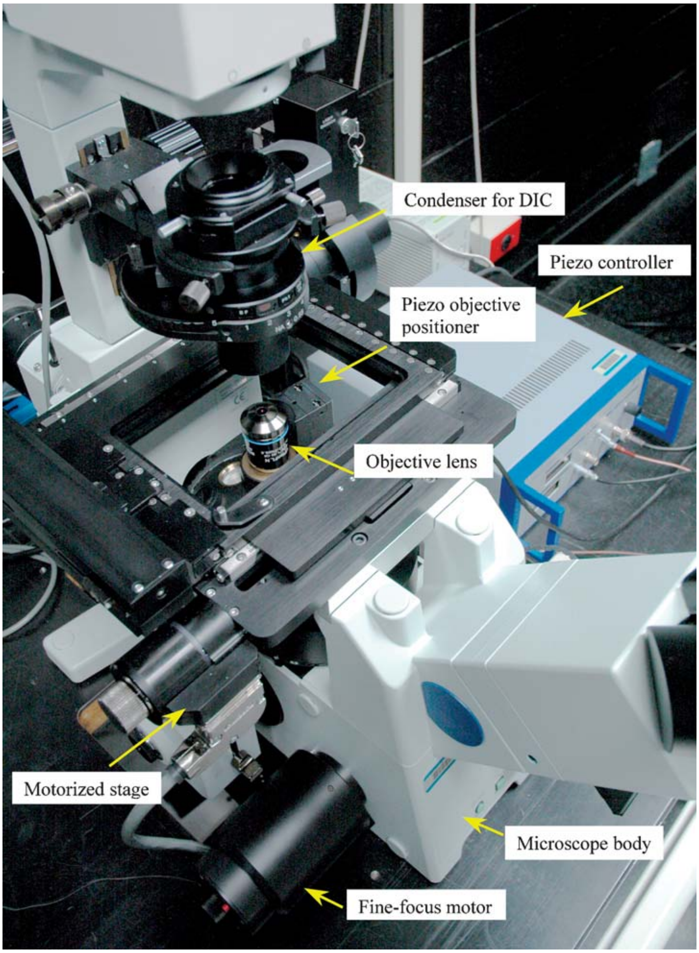

FIG. 1. The autofocus module on a microscope. The camera is mounted on the bottom port and is hidden from view.

not have sufficient contrast to allow autofocus because most live specimens are transparent and cannot be stained. Although phase-contrast imaging produces good contrast, phase objective lenses have a phase plate in the back focal plane that attenuates light when acquiring fluorescence images. A DIC-enabled microscope is ideal for our method because it both provides sufficient contrast and brings more high-frequency content in the spatial domain of the image.

For micropositioning the objective lens in its axial, or z position, various solutions are possible. We have used a piezo-electric transducer that attaches to the objective turret of the microscope below the objective lens. This Physik Instrumente P/N 7-721.10 PIFOC (Auburn, MA) had a limited travel range of 100 μm but an excellent theoretical precision of 24.4 nm. A Model P-725 PIFOC can also be used for a longer range of 400 μm with the same resolution. The PIFOC are controlled by an E-662 LVPZT amplifier with built-in servo feedback control. A better alternative is a piezo-stage insert that can be used to directly move the specimen for focusing. This insert is preferable because the conventional PIFOC is screwed onto the objective turret of the microscope. There it elevates the objective lens 13 mm from the original position, significantly degrading DIC images because of the additional space between the objective lens and the Nomarski prism.

An alternative to the PIFOC is an electric stepper motor attachment with linear-encoded feedback (Ludl Electronic Products Ltd., Hawthorne, NY: P/N 99A420 and 99A404 for Olympus IX71 microscope), installed on the fine focus knob of the microscope. Using this motor preserves DIC image quality. However, the linear encoder feedback precision is only 50 to 100 nm. The advantage of this focus z motor is the much longer range of motion (the entire range of the original focus travel of the microscope). One can use both the PIFOC and the z-axis motor on the same system for a microscope that has both the ideal range of the motor and the higher resolution and good repeatability of the PIFOC.

The motorized stage is used to translate the specimen in the x–y plane for scanning. We use a stepper motor stage from Ludl Electronic Products Inc. (P/N 99S108-02) for the inverted microscope. The stage has a linear encoder of 100-nm resolution for each axis and is capable of attaching a piezo-electric transducer add-on for vertical travel of the specimen. The stage motors are driven by a National Instruments MID-7604 power drive (National Instruments Corp., Austin, TX) and are interfaced with the computer.

Autofocus calculations are carried out by software on the computer. Therefore, focus speed increases as computer processing speed increases. Advances in microcomputers have made them powerful enough to carry out the calculation without resorting to the dedicated image processor hardware used in the past (Price and Gough, 1994). We use a Pentium Xeon dual 2.4-GHz CPU computer (model Precision 650, Dell, Round Rock, TX). During a scan, many processes are carried out by the computer, such as motion control of the stage and piezo device, timing, and computation of focus variables. Therefore, a high CPU clock speed is preferred. Parallelism such as multiple CPUs, dual-core CPUs, and Hyperthreading (HT) technology can provide an additional edge in multithreading these operations.

As other cameras and modules of the microscope require expansion slots, the host computer should be selected to have as many built-in PCI or PCI-X slots as possible. Our computer uses a PCI expansion slot on its main board for the image acquisition board. It uses another PCI expansion slot for an interface board (National Instruments Corp., Model PCI-7358) for the motor power drive. The same board also performs 16-bit digital-to-analog (D/A) conversion to control the piezo device by outputting a 0- to 10-V direct current (DC) voltage. The computer also has two IEEE 1394a ports for additional digital cameras.

The digital camera is interfaced with the computer to capture images for autofocus. The camera should have high bit depth, for example, 12-bit or greater, and be used for both autofocus and subsequent fluorescence image acquisition. In our system we have a Photometrics CoolSNAP ES monochrome camera (Roper Scientific, Inc., Tucson, AZ). This is a scientific grade camera that has 1392 × 1040 pixels in the CCD array. It has a fast 12-bit digitization rate of 20 MHz. Interfacing with the computer, we have a PVCAM PCI image grabber board (Roper Scientific, Inc). This camera coupled with the image grabber board can perform fast region-of-interest (ROI) acquisition for focusing on a specified region in the FOV. It also has an electronic shutter instead of a mechanical shutter for faster repeated digital autofocus and integrated fluorescent image acquisition.

Software

The software carries out the autofocus calculation instructions. It also controls the hardware, that is, stage movement, piezo movement, light path shuttering, and image acquisition via the camera. These controls utilize library functions from the motion control and image acquisition software development kits (SDKs) provided by the hardware manufacturers (Roper Scientific and National Instruments Corp.).

The autofocus method is based on image spatial resolution. When an in-focus image becomes out of focus, its two-dimensional spatial resolution decreases. The loss of resolution is manifested in decreased high-frequency content of the out-of-focus image. Our autofocus routine measures the high-frequency content at each plane of the image to determine optimal focus. High-frequency information is first separated from the full frequency spectrum of the image by applying a high-pass or band-pass filter. The filters can be implemented either with analog electronic devices or digitally with a computer. Digital filters are easier to implement than manufacturing electronic hardware boards (Bravo-Zanoguera *et al.*, 1998). C/C++ libraries such as Intel Integrated Performance Primitive 5.0 (Intel Inc., Santa Clara, CA) and IPL 98 (University of Denmark, Denmark) can be used to perform the filtering and calculations. The downloadable libraries are available on the Internet. High-level programming languages such as LabView 7.0 (National Instruments Corp.) and MATLAB 7.0 (Mathworks Inc., Natick, MA) are easier to use, but slower.

For each field, the micropositioner incrementally changes the distance between the specimen and the objective lens to search for the best focus position. An image is acquired into memory for each axial z position. Digital image processing is performed on each image to obtain its "focus index." A set of focus indices is then computed for each focus attempt, and these are fit to a curve to find the optimum focus. The focus measurement function used to compute the focus index for an image is

$$F(z) = \frac{\sum_x \sum_y [f(x,y) \otimes i_z(x,y)]^2}{\left[\sum_x \sum_y i_z(x,y)\right]^2}, \tag{1}$$

where (x, y) is the discrete coordinate in the image and $i_z(x,y)$ is the image pixel intensity acquired at each focus position, z, and $f(x,y)$ is a high-pass or band-pass filter. $\otimes$ stands for the convolution operator.

There are different choices for the digital filter, $f(x, y)$. Integer filter kernels are faster to compute than floating point. Custom-designed kernels usually do not use integers. Simple integer filters such as the one-dimensional derivative filter

$$[-1 \quad 0 \quad 1]$$

and the basic high-pass filter

$$\begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

can be sufficient (Oliva *et al.*, 1999; Price and Gough, 1994). Specialized, custom-designed filters such as the 31-tap floating-point kernel in the Appendix can also be used to fight contrast reversal (Oliva *et al.*, 1999).

By going through each axial position in the specimen, a set of focus indices is obtained to calculate the best focus position by fitting a curve over the indices, as shown in Fig. 2A. Thin specimens produce an ideal curve that can be fit readily to a Gaussian or polynomial function. However, when thicker specimens have discernible objects in different planes, multiple peaks can be produced in the curve, as shown in Fig. 2B. This is more frequently a problem in images taken using high NA lenses because the DOF is much less than the cell thickness. In multimodal curves, the

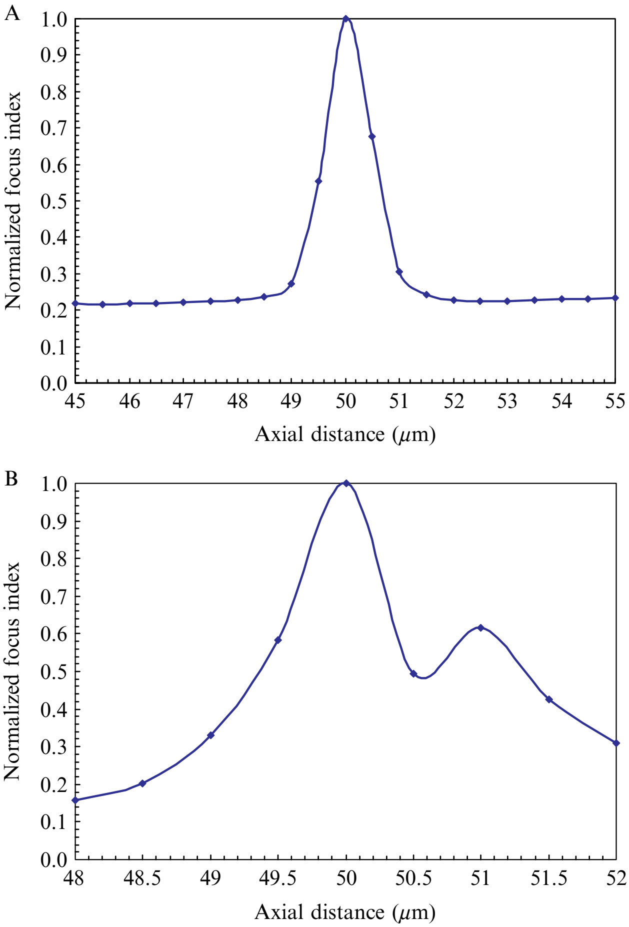

FIG. 2. Focus function curves from fixed fibroblasts. A curve from a thin specimen (A) is contrasted with that from a thicker specimen (B) in which more than one plane of focus provides discernible contrast. (Image a: 21 axial positions in the 10-μm search range, objective = Olympus PlanApo N TIRFM 60×/1.45 NA. Image b: 9 axial positions in the 4-μm search range, objective = Olympus UPlanFL N 40×/1.30 NA. Focus indices are normalized to one and connected using a smoothed line.)

highest peak represents the intracellular layer that contains the most detail. Curve fitting such data using a predefined function will fail. A weighted-average approach must be applied to counter this problem (Price and Gough, 1994).

The best focus position with power-weighted average is calculated as

$$B = \frac{\sum zF(z)^m}{\sum F(z)}, \tag{2}$$

where $F(z)$ is shown in Eq. (1) and m is an integer. The flowchart in Fig. 3 summarizes the process of finding the best focus z position. C++ code for the class member function "incremFocus" is listed in the Appendix.

When focusing a FOV, the in-focus position of the previous field is taken to be the middle axial position, on the assumption that the substrate supporting the cells does not change height abruptly. The very first field must be focused manually to set the starting in-focus point within the set focus range. The focus region of interest (ROI) is set and memory is allocated. The specimen is moved to the bottom of the focus range. From bottom up, a "for" loop is carried out to obtain a sequence of focus indices. In each step through the axial sampling, an image with pixel values of data type "int" is acquired. The pixel values must be scaled down to an 8-bit integer in order to prevent data overflow in calculations of the summations in Eq. (1). [Depending on the operating system, an "int" data type has different bit depths. For Windows 2000 and Windows XP, it is 32 bits (4 bytes) and has a range of 2^{32}. If 10- or 12-bit integers are used, the numerator and denominator in Eq. (1) can be greater than 2^{32} and cause computation errors.] The total illumination of the image is computed by summing all pixel grayscale values. Then the image is convolved with a filter kernel representing the high-pass or band-pass filter. The sum of the squared values of each element is then calculated for the convolution output array. Finally, the resultant sum is divided by the square of the total illumination to give the focus index. After all the planes are covered in the axial direction, memory for the image and its convolved outputs can be deallocated and a power-weighted average [Eq. (2)] is used to interpolate for the best focus position.

Viscoelasticity

When an oil or water immersion lens is used, the immersion medium between the specimen and the objective lens can hinder autofocus. This is because the specimen, the immersion medium, and the objective lens form an assembly that has prominent viscoelastic properties. When the objective lens is suddenly moved toward the specimen, the immersion medium is

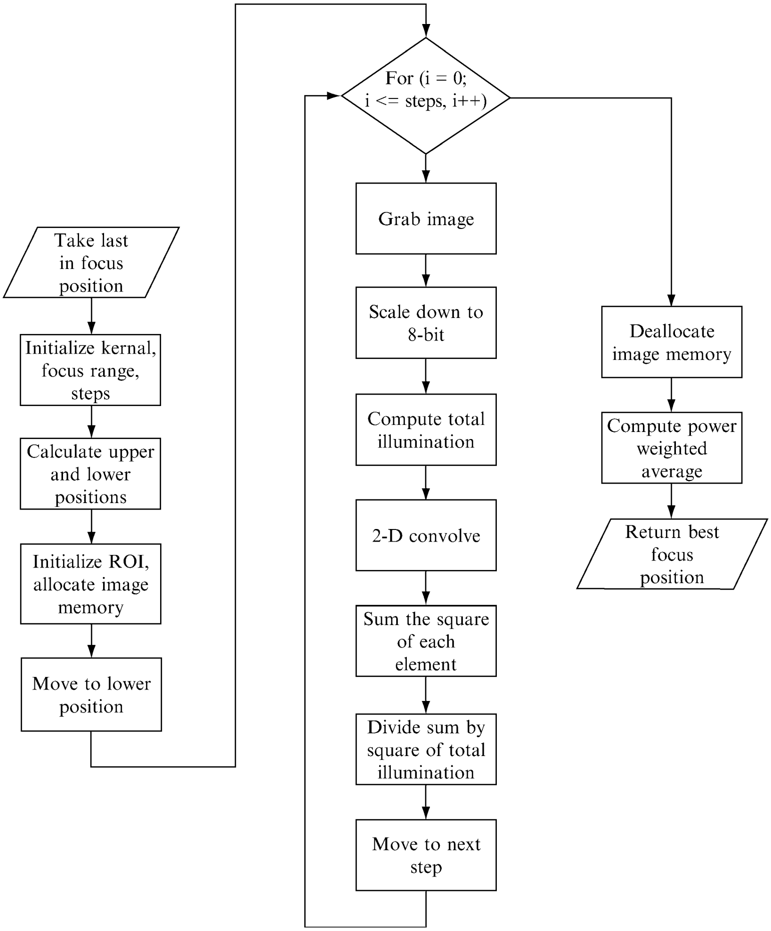
For (i = 0;
i <= steps, i++)
Take last in focus position
Initialize kernal, focus range, steps
Calculate upper and lower positions
Initialize ROI, allocate image memory
Move to lower position
Grab image
Scale down to 8-bit
Compute total illumination
2-D convolve
Sum the square of each element
Divide sum by square of total illumination
Move to next step
Deallocate image memory
Compute power weighted average
Return best focus position

FIG. 3. Flowchart of autofocus. This flowchart describes the algorithm for carrying out digital autofocus of a single field. The function in the Appendix shows an example of the implementation. It can be called repeatedly to perform autofocus of all fields in a scan.

pushed toward the specimen. Because the medium is incompressible, the specimen is pushed away from the objective and the spring mechanism in the objective is pushed away from the specimen. The specimen and the objective spring are elastic and yield as the medium flows to the sides, and the system reaches a new state of equilibrium. In the case of an air objective lens, the viscosity of the medium (air) is so little that the speed of reaching equilibration is limited only by the hardware mechanics. When oil is present, its viscosity dictates the focus speed. Therefore, image acquisition must be delayed after each step during the focus attempt for reliable focusing. Different immersion oils can be used so the user can opt to maximize speed (using low viscosity) or resolution (higher viscosity). The fluorescence of the oil should be checked at the wavelengths of interest so it does not interfere with the experiments. We use Olympus immersion oil (Olympus America, Inc., Melville, NY), which has low viscosity (135 cStoke) at 23° and an index of refraction of 1.516 that matches normal glass coverslips. Using this oil in our system, we set a delay of 25 to 50 ms to achieve fast autofocus (<0.8 s per FOV) with minimal autofluorescence.

Multiple-Field Scans

The automated microscope can perform a time-lapse scan of either a predesignated area encompassing multiple adjacent fields (area scan) or a set of predesignated fields of interest that may be separated from one another (point scan). Different applications may also call for obtaining several images at each FOV before moving to the next FOV (time priority) or obtaining a single image at each FOV while cycling between fields (space priority). The autofocus search range will be different for each of these imaging modes. In an area scan, the specimen support can be assumed to be flat enough so that heights of consecutive fields vary little. Therefore, the best focus position calculated from the previous FOV can be used as the midpoint z position in the autofocus search range of the current field. Steps 1 to 3 in Fig. 3 describe this method. For a 40× 1.30 NA lens, the autofocus search range can be as little as 1 μm. In a point scan, in-focus heights of different x–y locations may vary widely depending on how far a field is away from the previous field. Therefore, either the search range should be increased to accommodate all the fields or the user should log the in-focus positions for all the fields before the scan starts. If oil-immersion medium is used, enough oil should be applied to the scanned area so that the surface does not run dry, despite the sloughing of the oil caused by scanning.

Conclusion

This chapter summarizes the components and algorithm required to assemble a fully software-based autofocus system for automated microscopy screening. All components are off the shelf and are readily available. Software that carries out the autofocus can be written in any language that supports computer hardware data input/output and can resemble that listed in the Appendix. As modern microcomputers have become faster, as well as less costly, digital autofocus has become a solution that can be implemented easily. The autofocus module setup outlined here will repeatably produce sharp focus, despite the perturbations that impact scanning during microscopy-based high-content screening.

Appendix

What follows is the core of the autofocus, a member function "incremFocus" of the "CAutofocus" class written in C++. It is an implementation of the flowchart shown in Fig. 3. National Instruments' "flexMotion" and Intel's Image Processing Library (IPL) v2.5 C libraries are used.

```
    double CAutofocus::incremFocus(CImageGrabber
&grabOne, double cur_pos, double range, short steps)
    {
    float H[31]={0.00701,
    -0.00120, 0.00185, -0.01265, -0.01211, 0.08346,
    -0.04688, -0.18633, 0.27488, 0.13864, -0.58840,
    0.22454, 0.66996, -0.74667, -0.30163, 1.00000,
    -0.30163, -0.74667, 0.66996, 0.22454, -0.58840,
    0.13864, 0.27488, -0.18633, -0.04688, 0.08346,
    -0.01211, -0.01265, 0.00185, -0.00120, 0.00701};
    //int H[9]={-1, -2, -1, -2, 12, -2, -1, -2, -1};
    //int H[3]={1, 0, -1};
    IplConvKernelFP* pIplConvKern;
    double *FI;
    FI = new double [steps+1];
    double normConv, normIllum, powered, upper_pos,
lower_pos, pos;
    double maxFI=0, numerator=0.0, denominator=0.0;
    IPLStatus sta;
    int status;
    /* setup focus step locations */
    upper_pos = cur_pos + range/2;
    lower_pos = upper_pos - range;
    pIplConvKern = iplCreateConvKernelFP(31,1,15,0,
H);
    /* init *********************/
    grabOne.SetAFParams600,600,8;// center region
of field of view with 1×1 binning, low intg, 1× gain.
    IplImage* pInImg = iplCreateImageHeader(1,0,
IPL_DEPTH_16U,''GRAY'',''G'',IPL_DATA_
ORDER_PIXEL,
  IPL_ORIGIN_TL,IPL_ALIGN_QWORD,grabOne.
imgWidth,grabOne.imgHeight,NULL,NULL,NULL,NULL);
  iplAllocateImage(pInImg,0,0);
  IplImage* pSImg = iplCreateImageHeader(1,0,
IPL_DEPTH_32F,''
GRAY'',''G'',IPL_DATA_ORDER_PIXEL,
  IPL_ORIGIN_TL,IPL_ALIGN_QWORD,grabOne.
imgWidth,grabOne.imgHeight,NULL,NULL,NULL,NULL);
  iplAllocateImageFP(pSImg,0,0);
  IplImage* pMImg = iplCreateImageHeader(1,0,
IPL_DEPTH_32F,
''GRAY'',''G'',IPL_DATA_ORDER_PIXEL,
  IPL_ORIGIN_TL,IPL_ALIGN_QWORD,grabOne.
imgWidth,grabOne.imgHeight,NULL,NULL,NULL,NULL);
  iplAllocateImageFP(pMImg,0,0);
  /* end init ******************/
  pos = lower_pos;
  status=flex_load_dac(BOARD_ID,DAC4,posToDstep
(pos),0xFF); // output to analog out
  Sleep(20);
  /* the incremental process to calculate focus
index for each z plane (step) */
  for (int i=0; i<=steps; i++)
  {
  pInImg->imageData = (char*)grabOne.Acquire;
  sta = iplScaleFP(pInImg, pMImg, 0, 4095); // scale
down from 16 bit to 8 bit
  normIllum = iplNorm(pMImg, NULL, IPL_L1);
  iplConvolve2DFP(pMImg, pSImg, &pIplConvKern,1,
IPL_SUM);
  iplSquare(pSImg, pMImg);
  normConv = iplNorm(pMImg, NULL, IPL_L1);
  FI[i]=normConv/(normIllum*normIllum);
  if (FI[i]>maxFI)
  maxFI=FI[i];
  pos = pos + range/(double)steps;
  status=flex_load_dac(BOARD_ID,DAC4,posToDstep
(pos),0xFF); // output to analog out
  Sleep(50);
  }
  iplDeallocate( pInImg, IPL_IMAGE_ALL );
  iplDeallocate( pMImg, IPL_IMAGE_ALL );
  iplDeallocate( pSImg, IPL_IMAGE_ALL );
  iplDeleteConvKernelFP( pIplConvKern );
  grabOne.Complete;
  /* power-weighted average for best focus position
interpolation */
  ofstream out(‘‘D:\\work\\focusin.txt’’);
  for (int i=0; i<=steps; i++)
  {
  FI[i]=FI[i]/maxFI;
  powered = pow(FI[i], 12);
  denominator += powered;
  numerator += i * powered;
  out << FI[i] << ‘‘\n’’;
  }
  out.close;
  Sleep(50);
  pos = lower_pos + (range/(double)steps) *
(numerator/denominator);
  status=flex_load_dac(BOARD_ID,DAC4,posToDstep
(pos),0xFF); // output to analog out
  Sleep(50);
  delete [] FI;
  FI=NULL;
  return pos;
  }
```

References

Bajaj, S., Welsh, J. B., Leif, R. C., and Price, J. H. (2000). Ultra-rare-event detection performance of a custom scanning cytometer on a model preparation of fetal nRBCs. *Cytometry* **39,** 285–294.

Bravo-Zanoguera, M., Massenbach, B., Kellner, A., and Price, J. (1998). High-performance autofocus circuit for biological microscopy. *Rev. Sci. Instr.* **69,** 3966–3977.

Inoue, S., and Spring, K. R. (1997). “Video Microscopy, the Fundamentals.” Plenum, New York.

Oliva, M. A., Bravo-Zanoguera, M., and Price, J. (1999). Filtering out contrast reversals for microscopy autofocus. *Appl. Opt.* **38,** 638–646.

Price, J. H., and Gough, D. A. (1994). Comparison of phase-contrast and fluorescence digital autofocus for scanning microscopy. *Cytometry* **16,** 283–297.

Shen, F., Hodgson, L., Rabinovich, A., Hahn, K., and Price, J. (2006). Functional proteometrics for cell migration. *Cytometry* **69,** 555–572.