

# A computational image analysis glossary for biologists

Adrienne H. K. Roeder<sup>1,2,\*</sup>, Alexandre Cunha<sup>2,3</sup>, Michael C. Burl<sup>4</sup> and Elliot M. Meyerowitz<sup>1,5</sup>

## Summary

Recent advances in biological imaging have resulted in an explosion in the quality and quantity of images obtained in a digital format. Developmental biologists are increasingly acquiring beautiful and complex images, thus creating vast image datasets. In the past, patterns in image data have been detected by the human eye. Larger datasets, however, necessitate high-throughput objective analysis tools to computationally extract quantitative information from the images. These tools have been developed in collaborations between biologists, computer scientists, mathematicians and physicists. In this Primer we present a glossary of image analysis terms to aid biologists and briefly discuss the importance of robust image analysis in developmental studies.

**Key words:** Segmentation, Filtering, Registration, Volume rendering, Morphological image processing, Image analysis

## Introduction

In studying the development of an organism, biologists often need to determine quantitative properties such as the number of cells in a tissue, the size and shape of the cells, which cells contact one another, and the lineage of cells through time. Biologists extract these properties from images of the developing organism. Although humans are naturally good at reading and interpreting these properties, manual annotation is time consuming and monotonous. Overall, these limitations prevent the analysis of large numbers of images in a robust fashion. By contrast, computers can partially or fully automate the detection of these features. Computers can process large datasets of high-quality images in a fraction of the time that it would take to do this manually, even when a semi-automatic procedure is applied. An automated approach generally achieves high-throughput and highly reproducible results, and, when tuned appropriately, can achieve accuracy that is comparable to, or higher than, that generated via hand annotation. The detection and identification of objects in images are important fields of computer vision and image processing, and productive collaborations have been established between biologists and vision experts and algorithm developers. However, developing the appropriate software requires close communication between collaborators to identify suitable methods for the given images, and the great diversity of image content has so far prevented the development of a ‘one size fits all’ method.

Here, we present an introductory glossary of terms used in the computational analysis of images written from a biological perspective. In a large field that is rapidly advancing, no list is complete, but we have chosen to include terms that appear most frequently in our discussions and those that we feel are essential to know before pursuing more advanced topics. Owing to space constraints we have had to omit many other terms that would be worthy of inclusion. Our goal is to facilitate communication between biologists and their collaborators and to provide biologists in need of image processing with the basic concepts; this glossary, in fact, grew out of a necessity to facilitate our own collaborations.

In the sections below, image analysis terms, along with related concepts from computational geometry, are categorized by purpose and presented in the order in which they are usually performed: imaging and acquisition, preprocessing, segmentation, morphological processing, registration and tracking, and postprocessing. One can conceive many different image processing workflows to achieve the same desired outcome, so there are usually many solutions for the same problem.

## A typical image analysis workflow

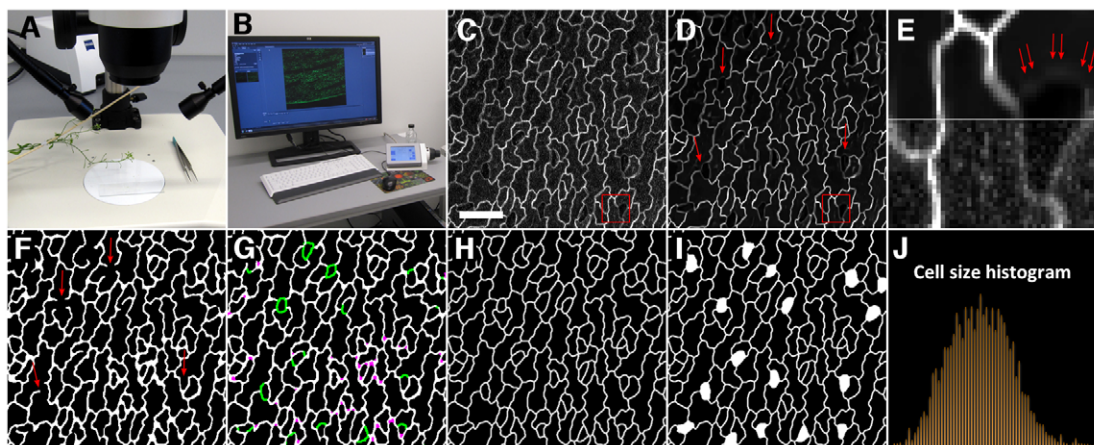
Generally, the first step in image analysis is to acquire high-quality images specifically optimized for image processing (Fig. 1A–C). For example, to measure cell sizes the biologist would perhaps image a tissue expressing a fluorescent protein targeted to the plasma membrane. In the image, the biologist would optimize the plasma membrane signal surrounding the cell while minimizing background fluorescence within the cell. Next, the images could be preprocessed to reduce noise and to enhance the contrast between features including membranes (Fig. 1D,E). Subsequently, the plasma membranes in the image could be automatically detected through one of the segmentation algorithms (Fig. 1F). Typically, several different segmentation approaches would be attempted and the parameters optimized for each dataset. The best one would be selected to detect the location of the plasma membranes. The detected plasma membranes can then be reduced with mathematical morphology to a skeleton (a single-pixel-wide outline) representing cell boundaries (Fig. 1H). The biologist can then check the validity of the automated processing by visually comparing the detected plasma membranes with the original image. At this point, it is typical that the biologist would intervene to correct errors in the segmentation (Fig. 1G–I). These errors are generally attributed to imperfections introduced at the imaging stage, such as breaks in the outline of a cell due to darker regions on the plasma membrane or the subdivision of cells owing to background fluorescence within the cell body, or are simply due to limitations of the algorithm in use. The cell sizes can then be quantified and presented as a histogram (Fig. 1J). Finally, the segmentation can be used to generate a triangular mesh to serve as the basis for a mechanical model simulation.

This is just one example of a possible workflow and, as mentioned above, a variety of workflows could be put to use depending on the desired outcome or the biological question in hand. Below, we discuss each of the workflow steps in turn, explaining the key terms and concepts that are applicable to each stage.

<sup>1</sup>Division of Biology, California Institute of Technology, Pasadena, CA 91125, USA.  
<sup>2</sup>Center for Integrative Study of Cell Regulation, California Institute of Technology, Pasadena, CA 91125, USA. <sup>3</sup>Center for Advanced Computing Research, California Institute of Technology, Pasadena, CA 91125, USA. <sup>4</sup>Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA. <sup>5</sup>Sainsbury Laboratory at Cambridge University, Cambridge CB2 1LR, UK.

\*Present address: Department of Plant Biology and Weill Institute for Cell and Molecular Biology, Cornell University, Ithaca, NY 14853, USA

†Author for correspondence (aroeder@cornell.edu)



**Fig. 1. Workflow example in image analysis.** Successful image analysis can be achieved when all steps from sample preparation to image segmentation are well integrated. Illustrated is an example of a processing pipeline for segmenting and measuring the cells present in the outermost cell layer of an *Arabidopsis thaliana* sepal. (A) The plant is carefully dissected with the aid of a microscope and tweezers. (B) A confocal stack is recorded and a maximum intensity projection of the plasma membrane (labeled with ML1::mCitrine-RC12A) of sepal cells is obtained using the microscope software. In subsequent panels, only a small portion of the green channel of the projection image is shown so that detailed comparisons can be made. (C,D) The original projection image (C), which is somewhat noisy, is filtered using the block matching 3D (BM3D) method to produce a denoised and contrast-enhanced image (D). The arrows in D and E indicate stomata boundaries that are barely seen after denoising; this is mostly due to the weak signal on stomatal membranes. (E) Magnification of the boxed region in C and D showing before (bottom) and after (top) denoising. (F) The plasma membranes obtained after a sequence of operations: (1) removal of any existing gross texture using a total variation-based filter; (2) localization of edges using a difference of Gaussians filter; (3) k-means thresholding (two regions); and (4) application of mathematical morphology operations to fill holes and remove tiny blobs (area opening and closing). The biologist is then prompted to assess and repair the results if necessary. (G) Manual corrections are performed: green indicates added edges and magenta indicates removed regions. (H) After corrections, an accurate segmentation comprising one-pixel-wide edges is achieved (blurred here to help visualization). (I,J) Stomata (white patches) are manually marked in the segmentation (I) in order to discard them in the analysis of cell area (J). Scale bar: 50  $\mu\text{m}$ .

## Imaging and data acquisition

The first step in any image analysis approach is to acquire digital images using a microscope or digital camera. Careful optimization of the settings during imaging can dramatically improve the success of subsequent computational image analysis. Image analysis can rarely extract good data from poor quality images. It is therefore useful to discuss the imaging protocol with the image analysis expert to optimize the images specifically for further processing, which can differ from optimizing them for human visual interpretation.

### Pixel and voxel

Short for picture element, a pixel is the basic two-dimensional (2D) unit from which a digital image is composed (Fig. 2A). Similarly, a voxel, short for volume element, is the basic 3D unit from which a digital volume is composed (Fig. 2B). Pixels and voxels have a defined position in their respective spatial Cartesian coordinate systems given, respectively, by two or three coordinates and associated color intensity values. For color images, multiple intensity values are associated with each pixel, one for each color channel. Pixels and voxels may have sizes (e.g.  $1 \times 1 \times 5 \mu\text{m}$ ) associated with the physical dimensions of the object being imaged.

### Stack

A stack comprises a set of related 2D images, or slices, collected from a single biological sample to form a 3D image. A z-series is a stack that is composed of a series of images spaced at precise intervals through the vertical (z) dimension of the sample (Fig. 2C). A time series is a collection of 2D or 3D images taken of the same sample over a time interval, revealing its time history. Confocal

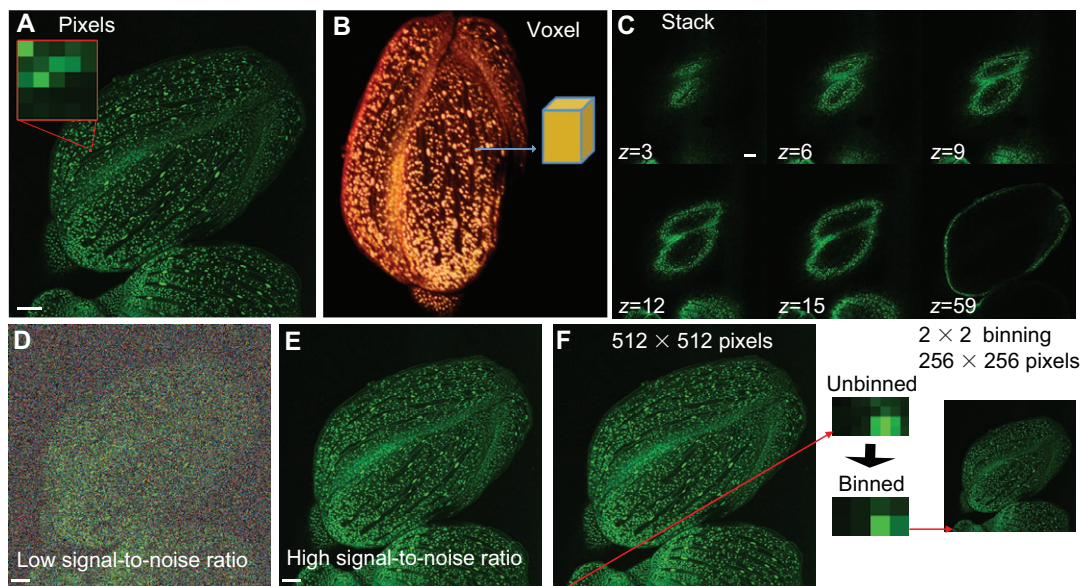
microscope software has functions for taking z-series and time series images. Biologists are currently generating 3D z-series stacks at many time points, resulting in 4D data.

### Sampling

The process of discretizing a continuous signal into a finite set of countable, discrete values is called sampling. The denser the sampling the closer the discrete signal is to the true, continuous signal it represents. Poor sampling leads to distortions of real observations. This is similar to the concept of sampling in statistics, where one works with samples as subsets of an entire population to represent it. Spatial sampling in digital imagery is the conversion of the physical space being imaged into a set of equally spaced grid points that represent the visible space. The finer the grid the greater the nuances one can capture from the real world. But one can only afford so much: grid resolution is constrained by available resources (e.g. digital storage, processing time, sensor capacity) and is dictated by the finest detail an optical device can read.

### Quantization and bit depth

Some fraction of the photons hitting the photosensitive detectors of the imager produce electrons, which are read out by the instrument electronics as digital numbers (DN values). The conversion from electrons to DN value is known as intensity quantization or light sampling. DN values represent the light intensity with a finite set of values. For color images, a separate DN value is produced for each color channel. The number of bits used to represent the intensity values of a pixel is called the bit depth; the larger the bit depth the more shades and colors one can represent but at the expense of using more computer memory. An 8-bit pixel, which is probably the most common type, allows for 256 ( $2^8$ ) intensities of



**Fig. 2. Image acquisition.** (A) A maximum intensity projection of a flower bud. Plasma membranes (ML1::mCitrine-RCI2A) and nuclei (ML1::H2B-YFP) are both fluorescently labeled (green). The inset shows individual pixels from the image. (B) A 3D (volume rendered) image of the flower bud shown in A. The inset represents a single voxel from the 3D image. (C) A confocal z-stack series of images through the flower bud shown in A and B. The interval between z-sections is  $4\mu\text{m}$ , and only a few representative sections are shown. (D,E) Low (D) and high (E) signal-to-noise ratio images of the flower bud. (F) A  $512 \times 512$  pixel image of the flower bud (as in A) is  $2 \times 2$  binned to make a  $256 \times 256$  pixel image. Four unbinned pixels in a square are averaged to create a single binned pixel. Scale bars:  $100\mu\text{m}$ .

gray, whereas a 16-bit pixel allows for 65,536 ( $2^{16}$ ) intensities. A 24-bit RGB image uses three 8-bit channels to code up to 16.7 million ( $2^{24}$ ) colors (but the eye cannot distinguish, nor can the computer produce, this many colors). Some microscopes use 12 bits per channel, giving 4096 ( $2^{12}$ ) possible gray values for every channel. In summary, in a B-bit image a single pixel can have up to  $2^B$  different intensity values.

### Image resolution

Image resolution is a general term used to express how fine an image is in space and intensity. Spatial resolution is related to spatial sampling and it is fundamentally the ability to resolve or distinguish small features in the image. In a high-resolution image, one can clearly identify fine detail. Intensity resolution refers to the ability to distinguish small changes in intensity values and is given by the image bit-depth. Note that although we can code up to 256 color values in an 8-bit image, in normal conditions we can only distinguish fewer than 150 on a computer screen. Image resolution depends both on the number of pixels recorded and the optics and electronic properties of the microscope (point spread function, signal-to-noise ratio, and sampling). Increasing the number of pixels while keeping the image size constant generally increases the resolution of the image; however, when properties of the microscope prevent the data in one pixel from being distinguished from its neighbor, additional pixels obtained via interpolation cannot increase the resolution.

Printing resolution is another aspect of image resolution. It is given by the number of dots per unit distance, e.g. dots per inch (dpi), used for displaying and printing an image. A given number of pixels can be displayed in different areas, which changes the dpi of the image. For example, a  $512 \times 512$  pixel image can be displayed at 72 dpi as a  $7.1 \times 7.1$  inch image or at 300 dpi as a  $1.7 \times 1.7$  inch image. If the same image were recoded at

$1024 \times 1024$  pixels, it could be displayed at 72 dpi as a  $14.2 \times 14.2$  inch image or at 300 dpi as a  $3.4 \times 3.4$  inch image. The key is to capture enough pixels at the time of image acquisition to resolve the features of interest.

### Noise

This refers to random perturbations in the intensity of an image that are not due to the object being imaged. Noise will always be present in any image, especially in low light conditions where the number of recorded photons is diminished compared with natural light conditions. Shot noise is random noise due to uncertainties when small numbers of photons are recorded. Speckle (also known as salt and pepper) noise is seen as extremely sparse random dots in the image caused by the electronic circuitry.

### Signal-to-noise ratio (SNR)

In imaging, the SNR is the ratio of the intensity of the object in the image (the signal) above the average background level divided by the standard deviation of the background, where the background consists of pixels not directly generated by the object (the noise). This ratio reflects how certain it is that a particular pixel in the image really represents an object in the image. The ability to detect objects reliably in an image therefore depends on its SNR. Images with high SNR are more prone to robust processing than images with low SNR, as it is easier to identify the objects in the image when the noise level is low (Fig. 2D,E).

### Binning

Binning an image down-samples the image by averaging blocks of pixels from the original image to create a single pixel in the new image. For example,  $2 \times 2$  binning of a  $512 \times 512$  pixel image would take every square of four pixels and average them to one pixel, resulting in a  $256 \times 256$  pixel image (Fig. 2F). This is a



fourfold decrease in image size. The purpose of binning is usually to reduce image size, but it can also improve SNR if the target objects are larger than a single pixel.

### Preprocessing

Preprocessing of images (i.e. filtering the image to reduce noise and increase its contrast) can improve the success of detecting objects in the image post-acquisition.

### Compression

Once the image has been acquired, image compression methods reduce the file size of digital images by filtering redundant information. Lossless image compression, such as the LZW compression scheme used in some TIFF images or the DEFLATE compression scheme used in PNG images, reduces the file size by encoding redundant patterns with shorter annotations that allow the exact image to be reconstructed from the compressed file. Lossy image compression applied, for example, in the JPEG format (with the exception of certain JPEG2000 settings), permanently removes some information from the image (Fig. 3A) and the original image cannot be recovered exactly after compression. Lossy compression schemes attempt to remove information from the image that does not compromise the viewing experience but might be problematic for image processing as they may disrupt the precise location of features in the image. Images are compressed by applying a transformation.

### Transformation

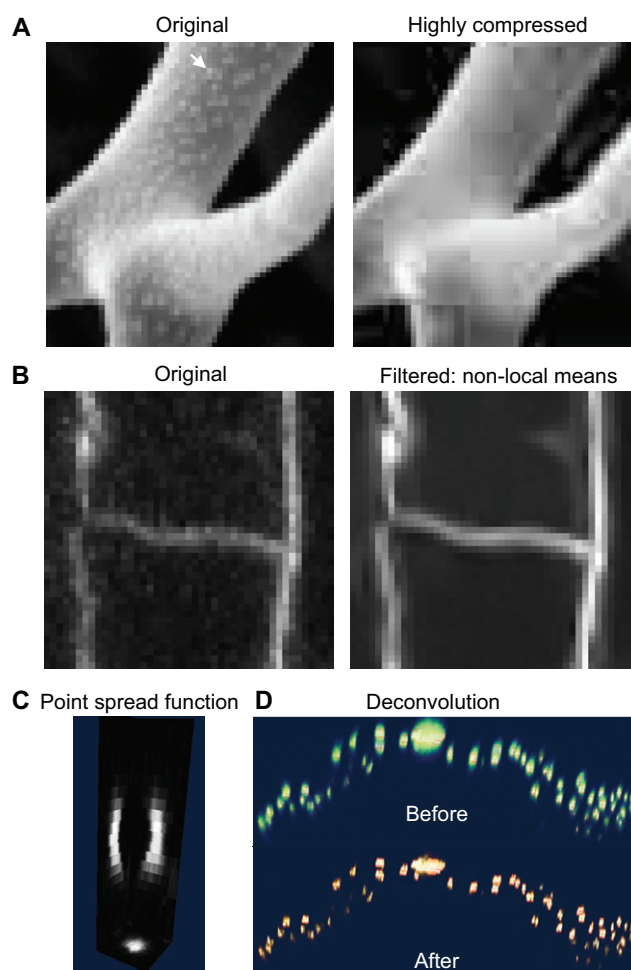
A process that applies a rule or formula to each pixel or set of pixels in an image in order to change that image in some way is referred to as a transformation. Many different kinds of transformations are applied in image processing for different purposes. For example, an affine transform can be applied during image registration to move and rotate one image such that it aligns to an image from a previous time point. A Fourier transformation can be applied to an image to detect edges or to quantify repetitive patterns in the image. Transformations such as the wavelet transform and the discrete cosine transform are also used for image compression, including the formation of JPEG images, which allows the images to be used and transmitted as smaller files.

### Fourier transform

The Fourier transform (FT) of an image provides a quantitative description of the texture and pattern in the image, which can be useful for image compression, reducing noise, detecting regions, and segmentation. The FT takes an image with pixels of specific brightness and  $x,y$  coordinates (spatial domain) and maps them to spatial frequencies (frequency domain). Specifically, the FT breaks an image down into a sum of trigonometric waves (sine or cosine). The magnitude of the resulting FT image emphasizes strong edges from the original image.

### Filtering

A filter is a type of transformation that modifies pixel intensity values. In neighborhood filtering, the pixel and its closest neighbors are combined to arrive at a new intensity value for the pixel. Filtering can be used to reduce noise (denoise), smooth, enhance or detect features of an image. For example, the median filter considers the neighboring pixels, sorts them to find the one with the median value and assigns that value to the pixel of interest. The mean filter, by contrast, averages the values of neighboring pixels and assigns that value to the pixel of interest. The simple Gaussian filter is often used



**Fig. 3. Preprocessing.** (A) Original scanning electron micrograph (left) of the branch point of an *Arabidopsis* trichome compared with the same image saved with high JPEG compression (right). Note that in the compressed image the cuticular dots (arrow) are lost. (B) Original confocal image (left) of fluorescently labeled plasma membranes (ML1::mCitrine-RC12A) of *Arabidopsis* sepal cells compared with the same image filtered with a non-local means filter to denoise the image (right). (C) Point spread function of the microscope showing the  $xz$  (left side),  $yz$  (right side) and  $xy$  (bottom) projections of a subresolution fluorescent bead. The point spread function shows the typical extension of fluorescence in the  $z$ -dimension from a single dot. (D) Cross-section images of nuclei before (top) and after (bottom) deconvolution. Note that the blurring of the nuclei, especially in the vertical dimension, is reduced by deconvolution.

for image smoothing, but more complex filters, such as the non-local means filter, can be applied to reduce noise while preserving texture patterns in the image (Fig. 3B). Filters that preserve the sharpness of edges in the image are superior as they keep object boundaries intact, thus facilitating detection and segmentation.

### Point spread function (PSF)

Ideally, a point source would generate a single point of light in the image. However, because light diffracts, scatters and spreads on its way from the object to the recording camera, one cannot precisely pinpoint the location of source points in the object. The PSF gives a likelihood region for a point source, generally in the form of an extended oblong (Fig. 3C). The PSF is used by deconvolution (see

**Table 1. Some image analysis tools and packages**

Package	Features and applications
Amira, Imaris and OpenDX	Software packages designed for 3D and 4D (time) visualization of imaging data as well as models. The software usually contains many useful image processing functions, including volume rendering, hand segmentation, registration, cropping in 3D and deconvolution. Amira, commercial software ( <a href="http://www.amira.com/">http://www.amira.com/</a> ); Imaris, commercial software ( <a href="http://www.bitplane.com/go/products/Imaris">http://www.bitplane.com/go/products/Imaris</a> ); and OpenDX, open source ( <a href="http://www.opendx.org/">http://www.opendx.org/</a> ).
CellProfiler	Open source image analysis software in which pipelines can be built to analyze large imaging datasets ( <a href="http://www.cellprofiler.org/">http://www.cellprofiler.org/</a> ).
Huygens Essential Image Processing	Commercial image processing software specialized for deconvolution ( <a href="http://www.biovis.com/huygens.htm">http://www.biovis.com/huygens.htm</a> ).
ImageJ/Fiji	Open source analysis programs that have specialized functions designed for biological image processing, such as performing measurements on calibrated images and handling confocal stacks. Plug-ins can add functionality, including many of the segmentation algorithms described (NIH, <a href="http://rsb.info.nih.gov/ij/">http://rsb.info.nih.gov/ij/</a> ; <a href="http://fiji.sc/wiki/index.php/Fiji">http://fiji.sc/wiki/index.php/Fiji</a> ; and <a href="http://rsb.info.nih.gov/ij/plugins/index.html">http://rsb.info.nih.gov/ij/plugins/index.html</a> ).
ITK, VTK and BioImageXD	Open source collections (libraries) of computer functions for the 3D visualization of images [Visualization Toolkit (VTK), <a href="http://www.vtk.org">http://www.vtk.org</a> ] and image segmentation and registration primarily designed for medical images [Insight Segmentation and Registration Toolkit (ITK), <a href="http://www.itk.org/">http://www.itk.org/</a> ]. These libraries have been implemented in software packages such as BioImageXD ( <a href="http://www.bioimaged.net/index.php">http://www.bioimaged.net/index.php</a> ).
KNIME	An open source data analysis platform that allows users to create analysis pipelines using icons, including preprocessing and segmentation ( <a href="http://knime.org/">http://knime.org/</a> ).
Loci Software	An open source collection of software developed for the life sciences. Of special interest is their Bio-formats program, which reads and writes many image file formats and is thus very useful for data conversion between formats ( <a href="http://loci.wisc.edu/software/home">http://loci.wisc.edu/software/home</a> ).
MATLAB or free alternatives such as Octave and Scilab	Programming language and numerical computing environment. New image processing tools can be developed within MATLAB and can be used in combination with the tools already available in the Image Processing Toolbox ( <a href="http://www.mathworks.com/products/matlab/">http://www.mathworks.com/products/matlab/</a> ; <a href="http://www.gnu.org/software/octave/">http://www.gnu.org/software/octave/</a> ; and <a href="http://www.scilab.org/">http://www.scilab.org/</a> ).
OMERO and Bisque	Open source software packages for image storage databases. Some functionality for visualization and analysis are included ( <a href="http://www.openmicroscopy.org/site">http://www.openmicroscopy.org/site</a> and <a href="http://www.bioimage.ucsb.edu/downloads/Bisque%20Database">http://www.bioimage.ucsb.edu/downloads/Bisque%20Database</a> ).
ParaView	Open source visualization software package for large datasets. It is built from VTK functions ( <a href="http://www.paraview.org/">http://www.paraview.org/</a> ).
Seg3D	Manual or algorithm-based segmentation of 3D volumes ( <a href="http://www.sci.utah.edu/cibc/software/42-seg3d.html">http://www.sci.utah.edu/cibc/software/42-seg3d.html</a> ).
R	A statistical computing environment. Images can be analyzed with the Image Processing and Analysis Toolkit for R ( <a href="http://www.r-project.org/">http://www.r-project.org/</a> ).

below) programs to reduce blurring from an image through attempting to return this oblong to a point. In practice, the PSF of a particular imaging configuration is obtained by imaging tiny fluorescent beads (e.g. those available from Molecular Probes or Polysciences). The deconvolution software attempts to estimate the PSF from the average image of many beads.

### Deconvolution

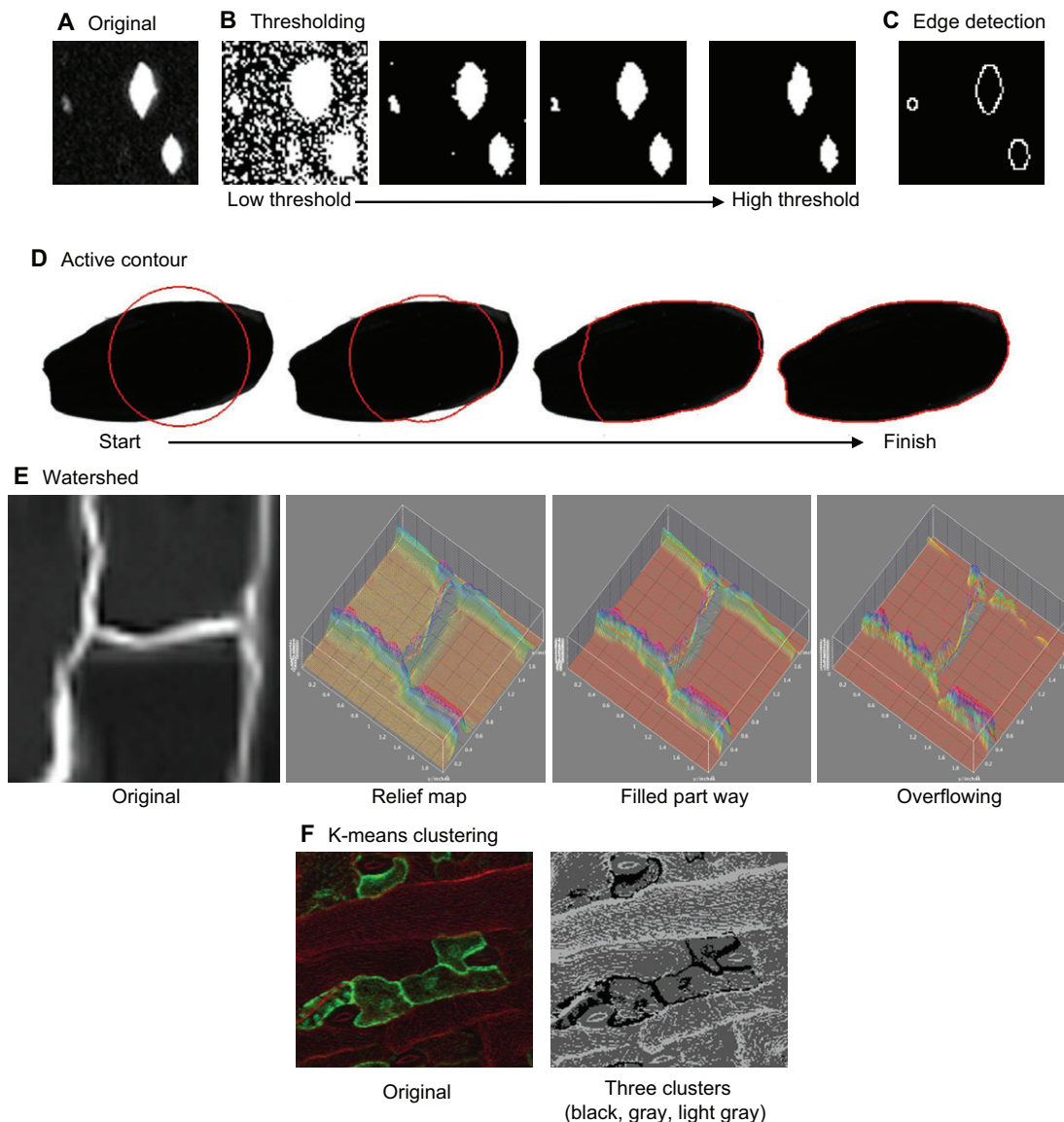
This is the computational process through which certain types of optical distortions, such as blurring, out of focus light and artifacts in an image, can be reduced (Fig. 3D). For example, some light emitted from a fluorophore in fluorescent images is observed in the *z*-planes above and below the plane in which the object is located, despite the absence of the fluorophore in those *z*-planes. Deconvolution algorithms computationally reduce this out-of-focus light based on the PSF of the microscope. Deconvolution is computationally difficult and works better on some images than others. Deconvolution packages can often be added to the software that comes with a microscope. Alternatively, 3D visualization software packages, such as Amira, can also perform deconvolution and it is a standard operation available in image processing packages and programs, including MATLAB and ImageJ (see Table 1 for a list of image analysis programs).

### Segmentation

The third step in image analysis often involves the automatic delineation of objects in the image for further analysis. Segmentation is the process of partitioning an image into homogeneous regions of interest and is often used to identify specific features (e.g. nuclei, plasma membranes, cells, tissue types). Segmentation is an active area of research with many different approaches. This section lists some of the methods available. In practice, it is often useful to start with a trial of several different methods to determine which works best for the particular set of images and features one wants to detect.

### Thresholding

One of the simplest methods for image segmentation involves thresholding so that every pixel above a defined intensity is assigned to one type of region, whereas every pixel below that intensity is assigned to another type of region. For example, all the nuclei brighter than a particular intensity can be detected by thresholding (Fig. 4A,B). Thresholding can be applied globally or locally to an image to account for differences in the intensities of pixels/voxels in different regions of the image. When adopted in an image processing pipeline, thresholding will produce masks that approximate the regions of interest.



**Fig. 4. Segmentation.** (A) Grayscale image of fluorescently marked nuclei from an *Arabidopsis* sepal. The nuclei are of different sizes owing to different DNA contents. (B) Thresholding segmentation to detect the nuclei shown in A; a low threshold includes a lot of background, but a high threshold loses the small nucleus on the left. (C) Edge detection detects the borders of the nuclei in shown in A. (D) Active contour segmentation of a stained *Arabidopsis* sepal to measure sepal size. The contour starts as a circle (red) placed over the sepal. The contour progressively bends and extends to match the border of the sepal. Note that segmentation of the sepal becomes much easier after staining to give it a uniform intensity. (E) Illustration of the watershed algorithm on the denoised images of the plasma membranes. The image is converted to a relief map based on the pixel intensity. As 'water' is progressively poured into the image, it fills the cells, showing the boundary until it overflows. (F) K-means clustering was used to detect different domains in this image, which shows an endoplasmic reticulum tagged with a GFP marker specific for certain cell types within the sepal (green) together with propidium iodide staining of the cell walls (red). The method was set to detect three clusters: the green GFP signal (black), the red propidium iodide staining of the cell walls (light gray) and the black background (a darker gray).

### Edge detection

This is a process in which edges (boundaries of objects) between regions of different intensity, color or another property are identified. For example, edge detection may be used to identify the outer boundary of each nucleus in an image (Fig. 4A,C). The most common methods of edge detection look for places with sharp changes in intensity. The image is thus considered as a relief map with peaks at the high intensities (e.g. the nuclei in Fig. 4A) and valleys at the low intensities (e.g. the background in Fig. 4A). The steepest slopes (places with high gradient magnitude) will be at the boundaries between the object (e.g. nucleus) and the background.

Although points where the slope is steepest (where the first derivative is maximized or minimized and the second derivative is close to zero) provide an estimate of where the boundaries of an object lie, additional steps are usually necessary to link edge pixels together into curves and to bridge across regions where the gradient information is weak.

### Active contours/snakes/balloons

Another approach for segmentation involves outlining objects in an image with single closed lines that develop according to an image minimization process (Fig. 4D). For example, in active contours



without edges, an initial circle or other set of closed contours is placed on the image, possibly covering each object or the collection of objects that one wants to segment. Forces are then exerted on the contour to bend, stretch, split, merge and contract it until it reaches the desired configuration, outlining the boundary of the object. In response to these forces, the contour moves and wiggles like a snake to seek the optimum position. One of the benefits of this method is that it can take prior knowledge about the region into account in the positioning and shape of the starting contour. The same concept can be extended into three dimensions using surfaces that expand like balloons.

### Watershed

A method developed within mathematical morphology to allow for segmentation of grayscale images. The image is considered as a relief map with peaks and ridges at the high, bright intensities and valleys at the low, dark intensities (Fig. 4E). Now imagine pouring water into a region of the image so that it progressively fills up one of the valleys, but does not spill into the next. The region filled with water is defined as a specific segmented object. For example, in an image of plasma membranes, imagine water being poured into the valley of the cell center and filling up the cell until it reaches the plasma membrane, thus marking an individual cell. The separate valleys are referred to as catchment basins and the ridges between the valleys as watersheds.

### K-means clustering

In image processing, k-means clustering (Fig. 4F) can be used to segment an image by partitioning the set of pixels into k clusters such that all pixels in a given cluster are similar in intensity, color, position or texture. In one version, the method is initiated with k central pixels, which can be chosen randomly from the set of initial pixels or based on some other initialization criteria. Next, the image is divided into k clusters each containing pixels most similar to their central pixel, one of the initially chosen k pixels. Then each cluster is averaged to determine its new central mean value, hence the name k-means. The process is repeated iteratively until none of the pixels changes regions after a new mean is computed. The advantage of k-means clustering is that it is computationally fast and available in many software packages. The disadvantages are that the number of clusters has to be chosen ahead of time (it can be an advantage if this is known a priori) and the quality of the result depends on the initial set of central points. The method only finds a local optimum and not necessarily the global optimum. In an image of fluorescently labeled cells, k-means clustering can be used, for example, to separate based on color the red cell walls and green endoplasmic reticulum regions (Fig. 4F).

### Gradient ascent/descent

An optimization method for finding local maxima or local minima in a function. It is sometimes used in image segmentation to identify points at local maxima and minima in the image intensities. Imagine that you wanted to identify the center of each nucleus in an image assuming that the nuclei are bright and the background is black. Start at a random point in the image and look at all the neighboring pixels. Move to the brightest neighboring pixel and repeat the process until the current pixel is brighter than any of the surrounding pixels. You will slowly move up the gradient of pixel intensity until you reach the local peak, which is likely to be the center of the nucleus. Although computationally inefficient, repeating the process starting from all the points in the image will identify all of the local maxima or centers of the nuclei. In the gradient ascent/descent method, not all of the nuclei need to be of the same intensity to be identified.

### Energy function

A function that, when evaluated for a specific set of input values, returns the cost of the system being modeled (usually the function is required to be continuous with continuous derivatives and bounded below). Energy functions are also called objective functions and are usually labeled H, E or J. The energy function should be constructed so that a solution that minimizes the energy provides a satisfactory solution to the 'real problem' that one is trying to solve. Frequently, gradient descent is used as a mechanism for finding parameter values that minimize the energy function, either locally or globally. Despite its name, the energy function does not necessarily have anything to do with energy in the conventional sense.

### Morphological image processing

Mathematical morphology is the branch of image processing and analysis that employs morphology as the key instrument to interpret and extract features from images. The idea is to structure the operations to be performed on an image according to the shapes of the objects that we are interested in detecting. It is often useful and sometimes necessary to adjust the shapes, sizes or width of segmented objects. For example, it might be useful to thin the thick edges representing plasma membranes to a single pixel width or to disconnect two nuclei that ended up merged in the segmentation because they are closely spaced. Morphological processing provides many tools to execute these tasks. The following section defines the basic terms and techniques that are used in morphological image analysis.

### Morphological processing

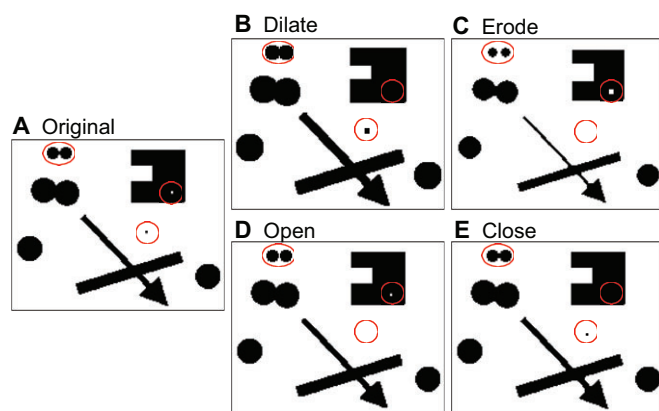
This processing involves changing the shapes of objects in binary (black and white) or grayscale images based on a set of morphological rules derived from mathematical set theory. Morphological operations can be used on images prior to or following segmentation, for example to separate objects that are wrongly connected. When counting points such as nuclei, morphology operations might be first applied to ensure the separation of otherwise connected nuclei. Morphological transformations can also be applied after segmentation to refine the results, for example by removing unwanted spurs, filling holes, removing tiny areas, or thinning lines. Image processing using mathematical morphology is conceptually simple and computationally fast. The drawback is that a set of operations for a particular image or classes of images normally has to be manually crafted for that specific image or class and might not apply to different images.

### Structuring element

This term refers to a shape used in morphological processing to determine which neighboring pixels or voxels should be considered when deciding the outcome for the central pixel under the shape. Circular disks, crosses, diamonds, lines and squares are frequently used as structuring elements. The structuring element is centered on a given pixel and only the neighboring pixels covered by the shape of this element are considered by the morphological operation being executed. The shape of a structuring element should reflect the shape of the target areas that one wants to modify.

### Dilation, erosion, opening and closing

These are the four most basic operations in mathematical morphology, each using a structuring element to produce its result. The image is analyzed pixel by pixel. Whenever any pixel of the structuring element intersects a foreground region, the pixel in the



**Fig. 5. Morphological image processing.** (A) A binary (black and white) image with geometric shapes serving as the starting point for morphological operations. Objects that change dramatically have been marked with red circles or ovals. (B) Dilation of the image in A showing that objects are enlarged. Note that the small hole in the back rectangle is filled. (C) Erosion of the image in A showing that objects shrink. Note that the small stray black pixel is lost and the two small dots are disconnected. (D) Opening of the image in A. Note that the small dots are separated and the stray pixel erased although the objects maintain their size. (E) Closing of the image in A. Note that the pixel is filled in the black rectangle, but the stray pixel remains and the small dots are attached.

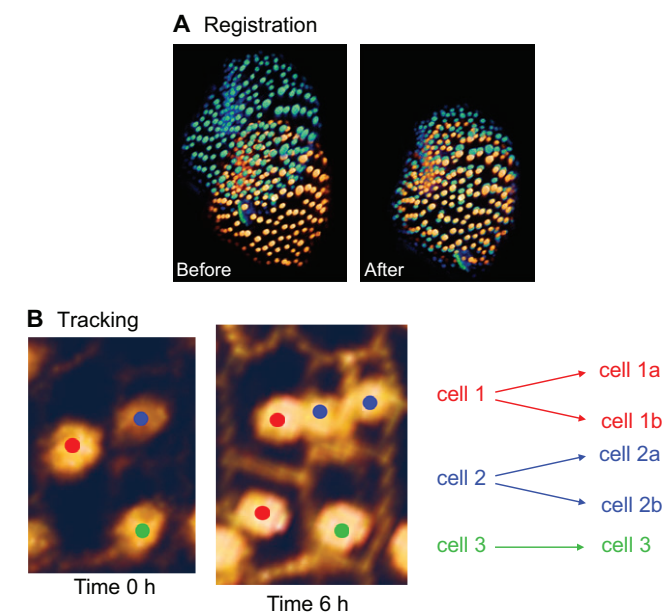
image located just under the central pixel in the structuring element is modified. Dilation is the operation in which an object is expanded (or broadened) through its borders (Fig. 5A,B). Dilation can be used to fill holes and close gaps in lines. Erosion is the operation for shrinking an object in the image and removing isolated pixels (Fig. 5A,C). Erosion can be used to disconnect attached objects by breaking thin bridges that may exist between the objects. For example, two nearby nuclei that are connected after initial thresholding segmentation can be separated by erosion. Opening is the application of erosion followed by dilation (Fig. 5D). Like dilation, it can be used to remove small objects. Closing is the application of dilation followed by erosion. It can be used to close small holes in an object (Fig. 5E). Like opening, it can be used to smooth the boundary edges of objects.

### Tracking and registration

For understanding development, it is particularly useful to take a series of images over time to observe changes in a developing organism. However, these time series images present additional challenges. The first is that the organism might move, so it is useful to align (register) the images relative to each other. Second, it is important to identify (track) the same cell across time points and through divisions. Below, we explain some key terms and approaches used during these processes.

#### Registration

In image analysis, this term refers to the process of aligning two or more images or volumes. Registration is often required to compare images of the same sample taken at different time points. Either the pixel intensity of the entire image or specific features within the image can be used to drive the alignment. Affine registration permits only global linear transformations (rotation, translation, scaling, shearing) of an image to align it with a reference image (Fig. 6A). By contrast, elastic registration allows for local warping of part of an image for better alignment to the reference image.



**Fig. 6. Registration and tracking.** (A) The panel on the left shows an overlay of images obtained from two time points during time-lapse imaging of a developing *Arabidopsis* sepal. The nuclei from the first time point are shown in gold and those of the second time point 6 hours later are in blue/green. After affine registration with Amira (right), corresponding nuclei nearly align, except for the changes that have taken place during the 6-hour interval. (B) To track the cells over time, corresponding sepal nuclei (gold) are identified at each time point. Plasma membranes are also in gold. At the first time point (0 hours, left), three cells are present and each is marked with a colored dot. Analysis and tracking of the cells 6 hours later (middle) reveals that the red cell has divided into two cells, both of which are marked with red dots. Similarly, the blue cell has divided, whereas the green cell has not. The cell lineages revealed by this tracking approach are shown on the right.

#### Thin plate spline

This is an approach to model displacements that are used in elastic registration for warping one image so that it aligns with another. Imagine that the first image is drawn on a flat, thin sheet of rubber and that certain key points (such as nuclei) are marked on the rubber. Now imagine bending that sheet so that the key points (landmarks) from the first image overlie the equivalent key points in the second image. The rubber minimizes the energy associated with bending. The points from the first image can thus be warped onto the second image. In addition, if the thin plate spline is being used to register images of a growing tissue at different time points, the amount of warping required to align the points is a measure of the change in shape or growth of the tissue.

#### Tracking

This refers to the identification of corresponding features in a set of images from a time series. For example, the nuclei might be tracked to produce the complete lineage of cells over time (Fig. 6B). Tracking generally requires prior segmentation and registration of the image. Difficulties in tracking usually arise when the cells being tracked have moved unpredictably, when they have divided such that one cell corresponds to two daughter cells in the next image, or when cells simply disappear in consecutive frames, for example when a fluorescent label is lost.



## Postprocessing and visualization

Once the images have been acquired and processed, they need to be presented in the most suitable format or represented in a manner that is accessible to the computer. These representations can then be used as the basis for mathematical modeling and further computation.

### Maximum intensity projection

To visualize 3D volume data, it is often convenient to project the volume data into a 2D image. Each pixel in a 2D image corresponds to a ray in 3D space. A single ray will intersect multiple voxels in the 3D volume. To decide which intensity to display in the image, the maximum voxel intensity along the ray is chosen, resulting in a maximum intensity projection (Fig. 7A). The capability to make a projection from a confocal z-series stack is usually provided with microscope software.

### Volume rendering

This process involves generating a 2D image from 3D data, such as a stack, so that the 2D image appears to be 3D on the computer screen (Fig. 7B). Ray tracing is one method for volume rendering. In this approach, rays of light from specified sources are traced through the image to determine how much light reaches a certain voxel and how much light is absorbed, reflected or emitted. The light rays are traced back to the viewer such that the resulting image is drawn from a single perspective. In many 3D visualization programs, the volume rendered image can be rotated and flipped such that all sides can be seen. Pixar's RenderMan software has been used for scientific rendering as well as for the popular children's movies.

### Mesh

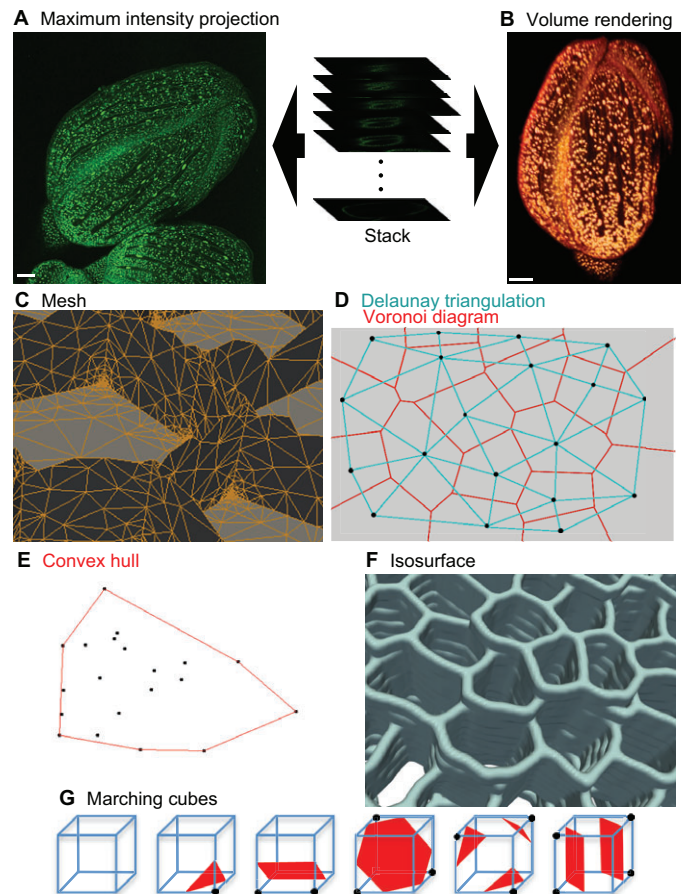
A mesh is a lattice of adjacent triangles (or other polygonal shapes, such as quadrilaterals) used to approximate a physical object in the computer (Fig. 7C). Surfaces in computer graphics are often represented as triangular meshes as they are computationally efficient to render. The triangles of the mesh can be small relative to the overall structure such that rounded surfaces (e.g. nuclei or plasma membranes) can be represented by small flat triangles with slight bends at their shared edges. The actual surface of the object determined by segmentation can be approximated using a mesh produced by triangulation methods such as the marching cubes algorithm (see below).

### Delaunay triangulation

A method for linking a set of points as the vertices of connected triangles (Fig. 7D). A Delaunay triangulation can use the set of points derived from image processing, such as points on the surface of the cell, to represent a surface of a structure, such as the plasma membrane. What distinguishes a Delaunay triangulation from other triangulations is the fact that the circle through the three vertices of each triangle does not include any of the other points in the set. This has some remarkable mathematical and computational properties, including the possibility, if adding extra vertices is allowed, of removing eventual skinny triangles that might be disruptive for simulations on computational surfaces.

### Voronoi diagram

A concept from geometry for partitioning space around a set of (generating) points (Fig. 7D). In technical terms, the Voronoi diagram of a given point set is a partition of the space into convex regions (called Voronoi regions), each surrounding a generating point, such that each point in the region is closer to its generating



**Fig. 7. Postprocessing.** (A) A z-stack of a developing *Arabidopsis* flower bud can be visualized through the formation of a maximum intensity projection. (B) Alternatively, the 3D data in the z-stack can be visualized by volume rendering. (C) A mesh (yellow) showing the structure of plant cell walls (gray). (D) The Delaunay triangulation (blue) and the Voronoi diagram (red) for a set of points (black). Note that the Voronoi divides the region into the points closest to each black dot. This approach is sometimes used to approximate cell boundaries around nuclei. (E) The 2D convex hull (red) surrounding the selected points (black). (F) Isosurface of the plant cell walls meshed in C. (G) Illustration of some of the possible cubes generated by the marching cubes algorithm. Black dots mark corners that are above the threshold, whereas unmarked corners are below the threshold. Based on the corners selected, surfaces are drawn through the cube. Individual cubes in the image connect to create a complete surface.

point than to any other generating point. The Voronoi diagram corresponds to the same partitioning of space produced by a nearest neighbor classification algorithm. It may sometimes be used to represent certain objects detected during the segmentation of an image. For example, a Voronoi diagram that uses nuclei centers as generating points can be used in certain circumstances as an approximation for the cell edges; each point in space is assigned to be part of the cell with the closest nucleus. Depending on the particular cellular system, the Voronoi cells may or may not accurately reflect the geometry of real cells.

### Convex hull

The smallest convex curve (2D) or surface (3D) (i.e. with no invaginations) that can enclose a set of points. In two dimensions, the convex hull around a set of points in the plane would be formed

by an imaginary rubber band stretched so that it just encloses all the given points (Fig. 7E). The line segment between any two points inside the convex hull must be entirely contained within the convex hull – this is the definition of convexity. The convex hull is a concept from mathematical analysis that can be applied after segmentation to determine, for example, the size of invaginations of cells.

### Isosurface

A surface in which all the points have the same constant value, for example a specified voxel intensity value (Fig. 7F). An isosurface of a zebrafish neuronal process, for example, would show the outlines of the fluorescently labeled dendrites.

### Marching cubes algorithm

A geometric method for generating a triangulated isosurface of a 3D image. Triangulations are used to visually render the objects they model in the computer. The surface is created at a user-specified pixel intensity value. For example, a brightness threshold might be used to create the surface of a fluorescently labeled endoplasmic reticulum. Imagine filling the volume of the image with cubes. For each cube, examine each of the eight corners to determine whether the corner is above or below the threshold. Color the corner black if it is above the user-specified intensity threshold or white if it is below the threshold (Fig. 7G). Cubes in which all corners are the same color are on the interior or exterior of the object and are ignored. The surface passes through cubes in which not all of the corners are the same color. Based on the color of the corners, the approximate surface can be drawn through the cube such that the white corners are on one side of the surface and the dark corners on the other. This surface is connected to the surface of the neighboring cube to generate an approximation of the surface of the object.

### Validation

After analysis, it is important to ensure that the image processing method is correctly detecting the desired features. The gold standard is to compare the automatically generated results of the algorithm with a dataset that has been manually annotated by the biologist. However, such manually annotated datasets are not common. At a minimum, the results of the segmentation should be overlaid on the original image and carefully compared for discrepancies. In our experience, there will be some errors or problems in the automatic segmentation, so it is useful for the biologist to be able to manually correct problems.

### Conclusions

Image processing is an incredibly powerful tool for quantitatively analyzing large image datasets when it is performed correctly. However, image analysis often requires testing different methods, developing new processing pipelines and carefully tuning the parameters for each image set. It is, after all, a craftsman's labor. It is thus important to validate the results by comparing them with the data. Computational image analysis is, nonetheless, extremely challenging because we still do not know how to translate the process of image interpretation, which is so natural to us as humans, into a computer program; what is obvious to a human is often not so apparent to a computer. For example, gaps in the signal of a plasma membrane are easily filled in by our visual system, but can completely disrupt the detection of cell boundaries even by the

most sophisticated algorithm. It is very difficult, if not impossible, for algorithms to take into account all of the possible variations and aberrations that can be encountered in images. Computer algorithms can be tweaked to consider small perturbations, but a minimal amount of human intervention can provide a more rapid and effective solution. Sometimes, a small adjustment in the imaging protocol can make subsequent analysis far easier. For example, to measure the size of an organ, detecting its outer edges is much easier if the organ has been stained in such a way as to present a unique and uniform color covering the entire organ. It is therefore crucial that biologists and image analysis experts work closely together to optimize all steps so that they can extract faithful quantitative information from image data. Despite these considerations, computer image analysis can provide a more complete understanding than is possible to achieve manually and is thus likely to be an integral part of future developmental biology studies.

### Further reading

**Gonzalez, R. C. and Woods, R. E.** (2007). *Digital Image Processing*, 3rd edn. Upper Saddle River, NJ, USA: Prentice Hall.

This is probably the most popular textbook in image processing and has been adopted for undergraduate teaching and widely used as a reference book by image processing developers.

**Lichtman, J. W. and Conchello, J.** (2005). Fluorescence microscopy. *Nat. Methods* **2**, 910-919.

A good introductory article aimed at biologists that explains the methods in fluorescence microscopy; of special interest is the itemized list of common factors that reduce the quality of fluorescence imaging.

**Ljosa, V. and Carpenter, A. E.** (2009). Introduction to the quantitative analysis of two-dimensional fluorescence microscopy images for cell-based screening. *PLoS Comput. Biol.* **5**, e1000603. An article that presents concepts and steps to perform in the analysis of fluorescence images; suitable for understanding the basics of image processing in fluorescence microscopy.

**Pawley, J. B.** (2006). Points, pixels, and gray levels: digitizing imaging data. In *The Handbook of Biological Confocal Microscopy*, 3rd edn (ed. J. B. Pawley), pp. 59-79. New York, NY, USA: Springer. Touches many of the definitions introduced in this glossary with longer and authoritative explanations of many aspects related to acquiring an image, using a microscope and the formation of digital images.

### Acknowledgements

We thank three anonymous reviewers, Kaoru Sugimoto, Xiaolan Zhang, Yun Zhou, Jill Harrison, Cory Tobin, Erich Schwarz, Henrik Jönsson, Tigran Bacarian, Eric Mjolsness, Heather Meyer and Marcus Heisler for discussions and comments on the manuscript.

### Funding

We acknowledge the Department of Energy Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences and Biosciences, Office of Basic Energy Sciences of the U.S. Department of Energy (E.M.M.) for funding the experimental research that has provided the images of sepals that are used in the examples given, the Gordon and Betty Moore Foundation Cell Center at Caltech (A.H.K.R. and A.C.) for funding the computational image processing of those images, and a Helen Hay Whitney Foundation postdoctoral fellowship to A.H.K.R.

### Competing interests statement

The authors declare no competing financial interests.