

**CellSegmenter: Installation and Usage**

John Aach  
May 23, 2007

## TABLE OF CONTENTS

Overview, Installation, and Requirements.....	3
Starting CellSegmenter .....	4
CellSegmenter Control Window.....	4
Opening an image and the CellDisplayer window .....	6
Manipulating thresholds; using False Color .....	8
Selecting and Removing Cell Segments .....	9
Using Barriers .....	11
Saving work done by CellSegmenter.....	13
Other CellSegmenter capabilities .....	15
Closing CellDisplayer and shutting down CellSegmenter.....	16
CellSegmenter .csm and .csf files .....	16
Citing CellSegmenter.....	17
Copyright .....	17

## Overview, Installation, and Requirements

CellSegmenter comprises a pair of MatLab 7.1+ GUI applications, each developed using the MatLab GUIDE facility. The name CellSegmenter is really the name of one of these applications, while the other is named CellDisplayer.

The CellSegmenter GUI is responsible for displaying and managing a user window that presents user controls for opening and saving image files, adjusting thresholds, selecting segments, and managing barriers. The CellDisplayer GUI is responsible for displaying and managing a window that displays a user-specified TIF file image field containing cell images, and supports user interactions with this image field such as mouse operations that select images or draw barriers. The two GUI applications intercommunicate to maintain the state of a user session.

The CellSegmenter and CellDisplayer GUI applications each comprise two components:

### *CellSegmenter*

- CellSegmenter.fig : MatLab .fig file that defines the window that presents user controls for loading, manipulating, and segmenting a cell image field.
- CellSegmenter.m : MatLab .m file that contains program logic for user interaction with CellSegmenter.fig control window

### *CellDisplayer*

- CellDisplayer.fig : MatLab .fig file that defines the window that displays a cell image field TIF file and enables user interaction with this image.
- CellDisplayer.m : MatLab .m file contains the program logic supporting user interaction with the CellDisplayer .fig file; i.e. that enables the user to interact with the cell image field

To install CellSegmenter, all four of these files must be downloaded and copied into the same directory.

For the analysis in our manuscript, CellSegmenter was coded, tested, and run exclusively on Windows XP machines using MatLab 7.1+ with the MatLab Image Processing Toolbox. Subsequent to this analysis, MatLab distributed release R2007a and CellSegmenter was modified to be compatible with it, and it is this modified version that we are distributing on our web site. We cannot provide assurances that it will run correctly on other operating systems or other versions of MatLab, as internals of MatLab GUI handling appear to be very release-dependent. Whatever release of MatLab you may use, the Image Processing Toolbox is required. CellSegmenter code is freely available under Harvard Technology Office license conditions specified on our web site <http://arep.med.harvard.edu/QMS> (choose the CellSegmenter program download hotlink). Users are welcome to adjust the source code if they wish to try running CellSegmenter in alternative environments.

***TIF images analyzed by CellSegmenter must be single-channel grayscale images with the suffix .TIF.*** Three-channel images generated by automated or manual microscopy must be preprocessed and reduced to a single channel prior to presenting them to CellSegmenter. Best results will be obtained with images that use the entire range of

intensities available to the decoder and in which intensity saturation is avoided to the degree possible.

## Starting CellSegmenter

Start MatLab and change the MatLab current directory to the directory containing the four CellSegmenter components. Then type in and enter "CellSegmenter" at the command line in the MatLab Command Window (see Figure 1). This action will bring up the CellSegmenter control window (Figure 2).

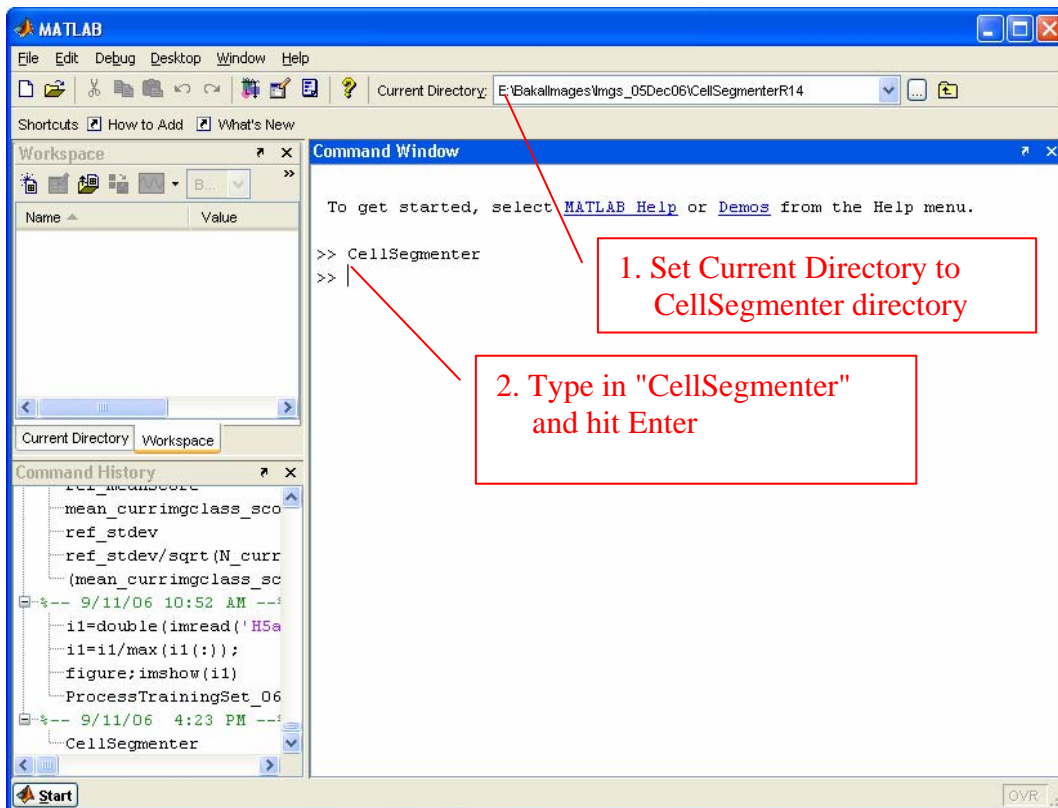


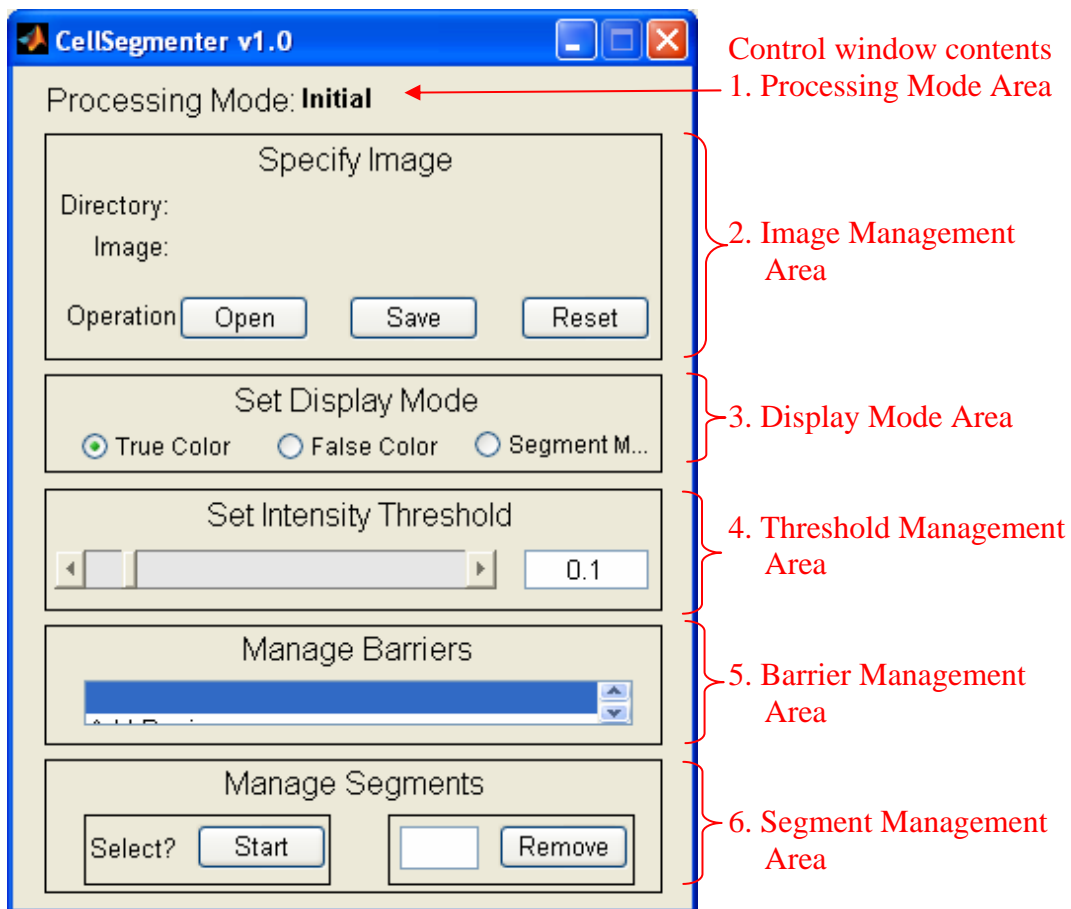
Figure 1: Starting CellSegmenter

## CellSegmenter Control Window

The CellSegmenter Control Window presents controls that allow the user to select and operate on a TIF image. Figure 2 displays the Control Window and identifies six sections within it.

1. The Processing Mode Area identifies the current processing mode. The processing mode provides information about the state of CellSegmenter and, in particular, how the next mouse operation in the CellDisplayer window will be interpreted. There 6 The Processing Modes are:

- Initial: CellSegmenter has just been opened and no image has yet been specified for processing. This is the mode shown in Figure 2. The next action must be to open an image.
- Display: An image is being displayed in a CellDisplayer window (see Figure 3), but mouse operations in this window will have no action.
- Segment Selection: CellSegmenter expects the user to point and click in the CellDisplayer and select a cell segment (see Figure ?)
- Manage Barrier: CellSegmenter expects the user to use the mouse operations to add or remove a barrier (see Figure ?)



**Figure 2: CellSegmenter control window and its contents**

2. The Image Management Area contains controls for opening, saving, and resetting images, and also displays the name of the current open image.
3. The Display Mode Area contains controls that specify the way a cell image field is to be displayed in the CellDisplayer window.

4. The Threshold Management Area contains controls for setting the current intensity threshold, which in turn defines borders around regions of higher intensity in an image that can be identified as cell segments.
5. The Barrier Management Area contains a list control that specifies what barrier operation may be executed -- adding a barrier, selecting a barrier, or deleting a barrier
6. The Segment Management Area contains controls that allow cell segments to be defined or removed.

## Opening an image and the CellDisplayer window

To open an image, click on the Open button in the Image Management Area (Figure 1). This will display a directory navigation area and file list window with the title "Select image TIF file" which you may use to specify the TIF file you wish to open. Only TIF files are displayed in the list. When you have specified your TIF file and clicked the Open button in the "Select image TIF file" window, the image will be loaded and displayed in a separate CellDisplayer window. The Image Management Area of the CellSegmenter window will now display the name and directory of the open imaged, and the Processing Mode will be Display. At this time, CellSegmenter only supports gray-scale TIF images.

If, when you open an image in CellSegmenter, there is no CellDisplayer window already open, a new CellDisplayer window will be opened. If, however, there was a CellDisplayer window at this time, CellSegmenter will reuse this window for the new image: You can never have more than one CellDisplayer window open at any one time. If you are reusing a CellDisplayer window that contains a prior image, and that image

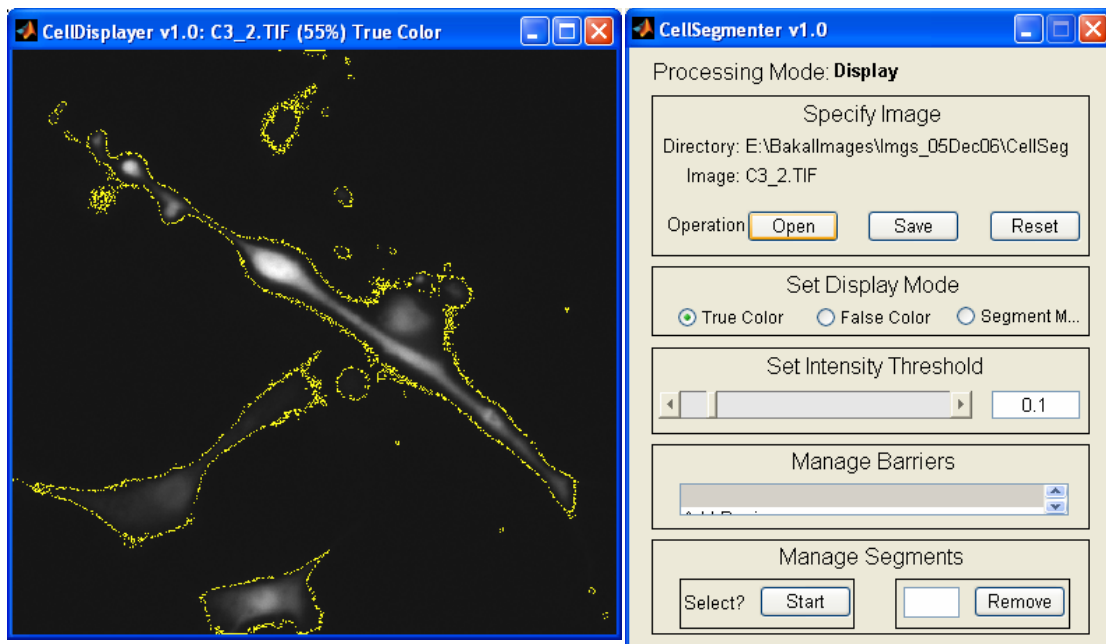


Figure 3: CellSegmenter and CellDisplayer after opening an image

contained changes that had not been saved prior to opening the new image, you will receive a pop-up dialog box asking you if you want to save the prior image prior to opening the new one.

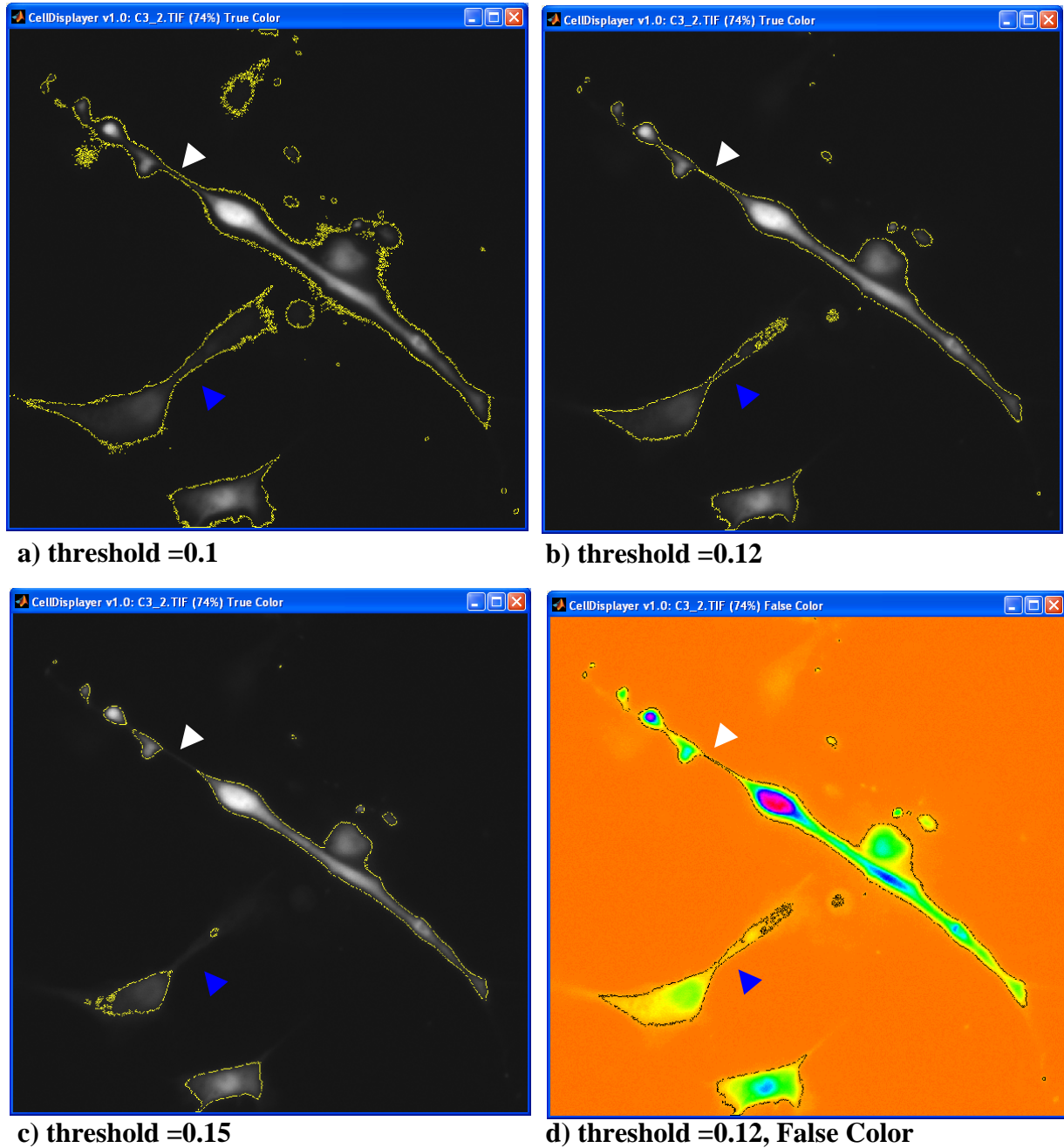
Figure 3 shows CellSegmenter after opening a window, along with the CellDisplayer window that contains the image. When a new CellDisplayer window is opened, it is possible that it will overlap the CellSegmenter window. These windows may be moved about your desktop by normal mouse operations on the title bar to eliminate or reduce the overlap (as in Figure 3).

Note that, in Figure 3, the CellSegmenter Image Management Area now contains the directory and name of the image that was just opened, and that the Processing Mode now reads Display. Meanwhile, the title bar of the CellDisplayer window in Figure 3 contains the expression "CellDisplayer v1.0: C3\_2.TIF (55%) True Color". This provides four pieces of information: the current version of CellDisplayer (v1.0), the file name of the TIF image (C3\_2.TIF -- this is also displayed in CellSegmenter), the size of the cell image displayed given as a percentage of 'actual size', and the current Display Mode (here, 'True Color' -- also indicated in the CellSegmenter Display Mode Area; more on this below). 'Actual size' represents the situation in which pixels in the TIF image and on your display are in a 1:1 relationship.

The CellDisplayer window may be resized by normal mouse operations (unlike the CellSegmenter window, which *cannot* be resized). When resizing, CellDisplayer always constrains horizontal and vertical resizing to the same rescaling factor, so that the cell image is never distorted. As you resize, the percentage of actual size shown in the CellDisplayer title bar changes accordingly. CellDisplayer resizing is useful because image displayed in CellDisplayer may have a pixel density that is higher than the density of pixels in the image, which results in loss of detail in the image displayed; this detail may be restored (at least partly) by enlarging the CellDisplayer window. When resizing, CellDisplayer always tries to keep the window within the bounds of your display. This may limit the extent to which you can enlarge the image.

The CellDisplayer window always displays the complete contents of the cell image field represented in the TIF image, although at greater or lesser scales. There is no way within CellSegmenter or CellDisplayer to display only subregions of the cell image field (except by the roundabout way of cropping the TIF file prior to opening in CellSegmenter). This is consistent with the purpose of CellSegmenter -- which is to allow a user to define and save particular cell segments within a cell image field, and not to support detailed image analysis or processing.

Along with the cell image field itself, you will see thin yellow lines running around regions of high intensity in the image field (see Figure 3). These are the borders around cell regions that are defined by the current setting of the Intensity Threshold in CellSegmenter. The yellow lines represent boundary pixels of thresholded regions in the image. These may appear spotty when the size of the CellDisplayer image is less than full size (i.e., when the title bar specifies a percentage size < 100%). They may also appear diffuse and noisy when the intensity threshold is low, resulting in the thresholded regions having noisy boundaries.



**Figure 4: Cell image field displayed in CellDisplayer with different thresholds. a-c) thresholds increase from 0.1 to 0.12 to 0.15; d) threshold 0.12 displayed in False Color**

## Manipulating thresholds; using False Color

Thresholds are manipulated by using controls within the CellSegmenter Threshold Management Area (Figure 1). Using the slider control on the left allows the user to raise or lower the current threshold using mouse operations. On the right, a text box gives a



numerical measure of the current threshold. The numerical threshold measure changes as you move the slider, and, conversely, the slider adjusts itself if you enter a new threshold in the text box. Numerical intensity thresholds are always normalized to the maximum intensity in the image, and so range between 0 and 1.

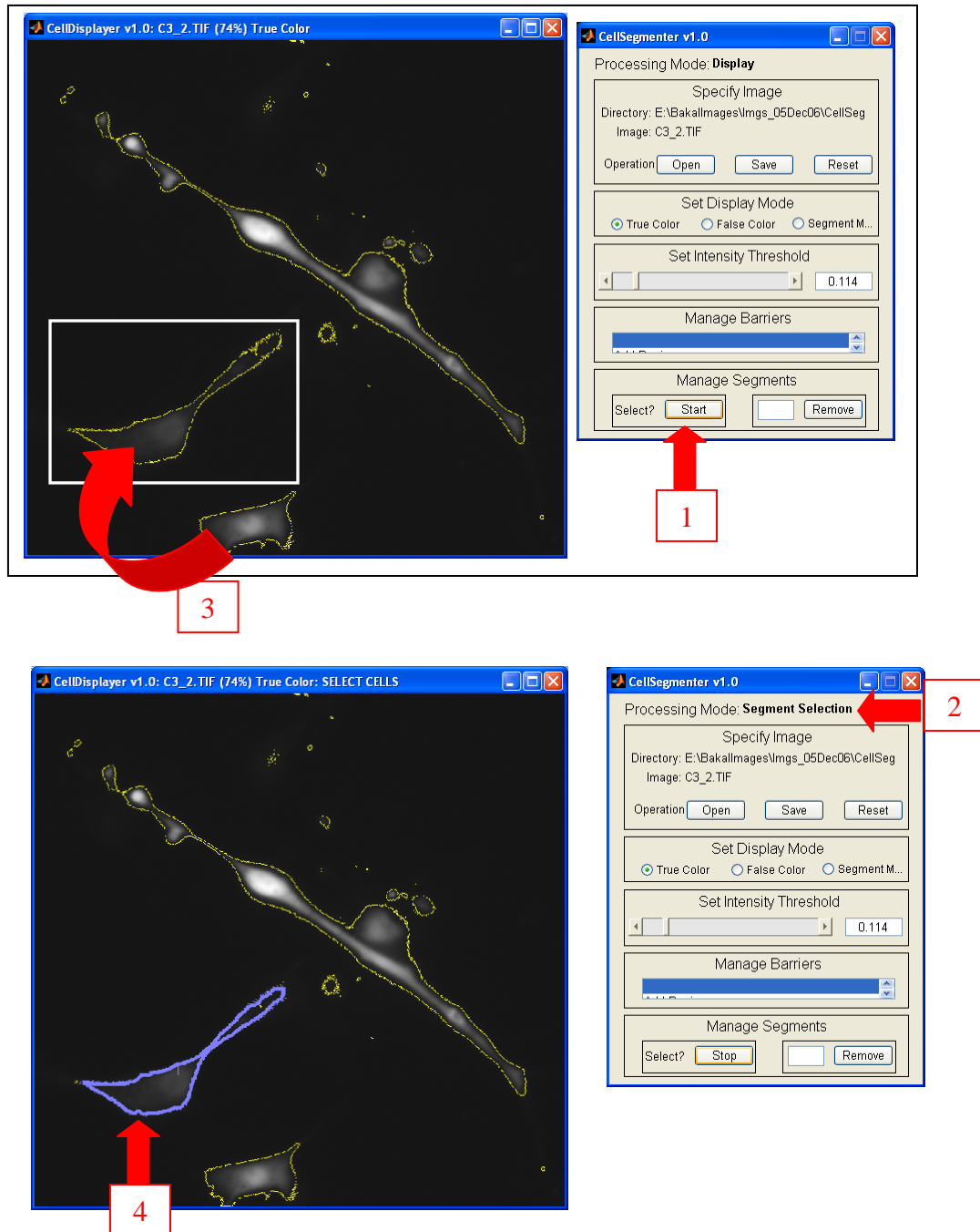
Figure 4 shows how the borders in a CellDisplayer image change as one adjusts the intensity threshold. As the intensity threshold increases in panels a)-c), the borders draw in closer to regions of higher intensity and may also become sharper and less diffuse as the threshold gets further and further above the intensity noise level. What may also happen is that single borders around large single regions that are greater than a low threshold intensity may split into borders around multiple, brighter intensity peaks as the threshold is raised; examples in Figure 4 are indicated with the white and blue arrowheads. At low intensity thresholds, individual cells in the cell image field will be merged within single borders, borders will be noisy and wide and extend far from visible cell boundaries, and borders may also form around noise areas that contain no cells. At high intensity thresholds, borders will cut off dimmer boundary regions of visible cells, and cells may be improperly fragmented as separate borders form around individual multiple peaks with a cell. It is up to the user to find an intensity threshold that, to the extent possible for a given cell, defines a border that most accurately tracks the visible cell boundary. It may be possible to find thresholds that define accurate borders for only a subset of cells in the image.

To assist the process of finding a accurate borders for a cell, the user may switch to False Color mode (see Figure 4d for a CellSegmenter False Color rendition of the image in Figure 4b). False Color renders gradations of low intensity in the cell image field more visible, enabling a better estimation of how threshold-defined borders correspond to actual cell boundaries for cells in the image. To switch to False Color Mode, click on the radio button labeled False Color in the Display Management Area of the CellSegmenter window. To switch back to True Color, the user simply clicks on the True Color radio button in this Area. In False Color mode, the threshold-defined borders are shown in black, vs. yellow for True Color. Note that since only gray-scale TIF images are supported by CellSegmenter are monochrome, True Color really means gray-scale.

## Selecting and Removing Cell Segments

Once an intensity threshold has been identified which draws an appropriate border around a cell, the region within this border may be defined as a "cell segment," i.e., a region of the image that represents the cell that will be saved and made available for further image analysis. The procedure for doing this is shown in Figure 5.

Starting with the CellSegmenter and CellDisplayer windows in the black rectangle at the top of Figure 5, let us suppose that the current intensity threshold has been judged to define an accurate border of the cell shown in the white rectangle. The first step, shown by the [1], is to click on the button in the Select? region of the Segment Management Area of the CellSegmenter window. This puts CellSegmenter into Segment Selection mode [2], which means that any left mouse click into the interior of a threshold-bordered region of the CellDisplayer window will cause CellDisplayer to define the bordered region as a segment. Thus, when the user left mouse clicks into border of the cell in the white rectangle ([3]), the selected region becomes a segment. CellSegmenter



**Figure 5: Selecting a cell segment (see text for details)**

acknowledges this action by highlighting this border and writing a segment number into the region ([4]). In this case, the segment number is 1.

As long as CellSegmenter remains in Segment Selection mode, the user may click into bordered regions of the CellDisplayer window and define additional segments. During this process, the user may adjust the intensity thresholds to use a different threshold for a different cell, and switch back and forth between True Color and False Color to check the

accuracy of the borders, prior to selecting additional segments. Changing the intensity threshold does not affect previously defined cell segments.

Segment Selection mode may be terminated by once again clicking the Select? button in the button in the Segment Management Area. Note that when Segment Selection mode is in effect, the label on this button changes from Start to Stop. (Segment Selection mode may also be terminated by switching to Manage Barrier mode; see below.) Notice also that in Segment Selection mode, the title bar of the CellDisplayer window contains the text SELECT CELLS in addition to the other information that is maintained there.

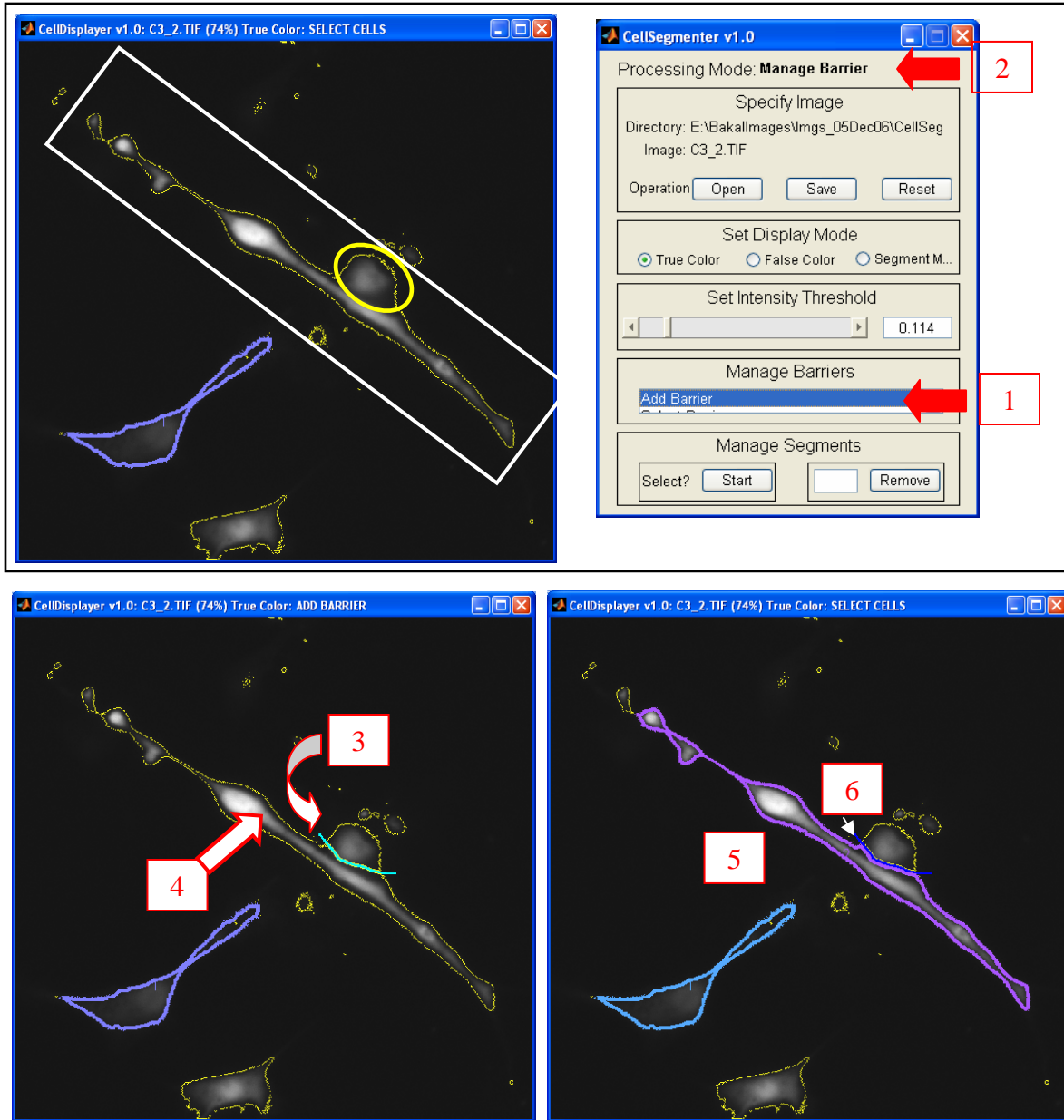
To remove a previously defined cell segment, the user simply enters the segment number into the text box on the right of the Segment Management Area and clicks on the Remove button. CellSegmenter will acknowledge this action by removing the highlighted border around the corresponding segment in CellDisplayer. Removing a cell segment may cause renumbering of remaining segments. Note also that a cell segment may be removed at any time, regardless of whether CellSegmenter is in Segment Processing mode. This is because the removal procedure does not require mouse actions in the CellDisplayer window, which need to be interpreted according to the current processing mode.

There is no explicit CellSegmenter operation for correcting or changing an existing cell segment. Any such correction or change must be performed by removing the segment and adding it anew.

## Using Barriers

Now suppose you want to continue selecting the segments in the image processed in Figure 5. You would like to select the elongated cell shown in the white rectangle in Figure 6, but this cell seems to be abutting a small round cell (yellow ellipse), and you cannot find a threshold that separates the elongated cell from its small round neighbor that does not seem to destroy the shape of the elongated cell. This situation may be handled by drawing a barrier. A CellSegmenter barrier is a hand-drawn linear contour that is incorporated into the boundaries defined by any threshold.

To draw the boundary, you select the Add Barrier item from the list box in the Barrier Management Area of the CellSegmenter window (shown as [1] in Figure 6). This puts CellSegmenter into Manage Barrier mode ([2]). In this mode, CellDisplayer expects the user to draw a linear contour by clicking and holding down the left button of the mouse on one endpoint of the intended barrier in the CellDisplayer window, then slowly moving the mouse cursor along the track of the intended barrier while keeping the left button down, and, finally, releasing the left mouse button at the other endpoint of the barrier ([3] in Figure 6). At the end of this operation, once the button is released, the barrier will be displayed as a thick white line ([3]).



**Figure 6: Adding a barrier and selecting a segment that incorporates it**

**Important Note:** Unfortunately, MatLab's GUI capabilities are not very efficient, so that the line is not drawn at the same time as it is being traced by the mouse. Attempts to implement this in development led to an unsatisfactory situation in which the CellDisplayer image was repeatedly displayed in full, rendering the drawing operation very slow and jerky. Because the contour must be drawn 'blind', the barrier may need to be removed and redrawn until it is right using the Remove Barrier operations described below.

With the barrier in [3], the elongated cell may now be selected as a cell segment without also getting the unwanted round cell, by clicking the Select? "Start" button in the Segment Management Area of CellSegmenter and clicking into the elongated cell in CellDisplayer (as described in the section "Selecting and Removing Cell Segments")

above). These *two* operations are represented in Figure 6 as [4]. (Note that these operations switch CellSegmenter out of Manage Barrier mode and into Manage Segment mode.) The elongated cell has become a cell segment ([5]) as desired. Meanwhile, the barrier remains displayed as a blue line ([6]).

When barriers need to be added for several cells, one way to proceed is a cell at a time according to the procedure just described, but this requires that one switch back and forth between Manage Barrier and Manage Segment modes for each cell. An alternative is to go into Manage Barrier mode by selecting Add Barrier as in [1] in Figure 6, and then adding *all* the barriers one needs in the image. When in Manage Barrier mode, while Add Barrier has been specified, CellDisplayer interprets each left-button-down / draw / left-button-up operation of the mouse in the CellDisplayer window as an order to define a barrier, so that all barriers may be drawn one after the other. Then, by clicking on the Select? Start button in the Segment Management Area of CellSegmenter, one can switch into Manage Segment mode, and click on all the CellDisplayer regions of interest to define the desired cell segments.

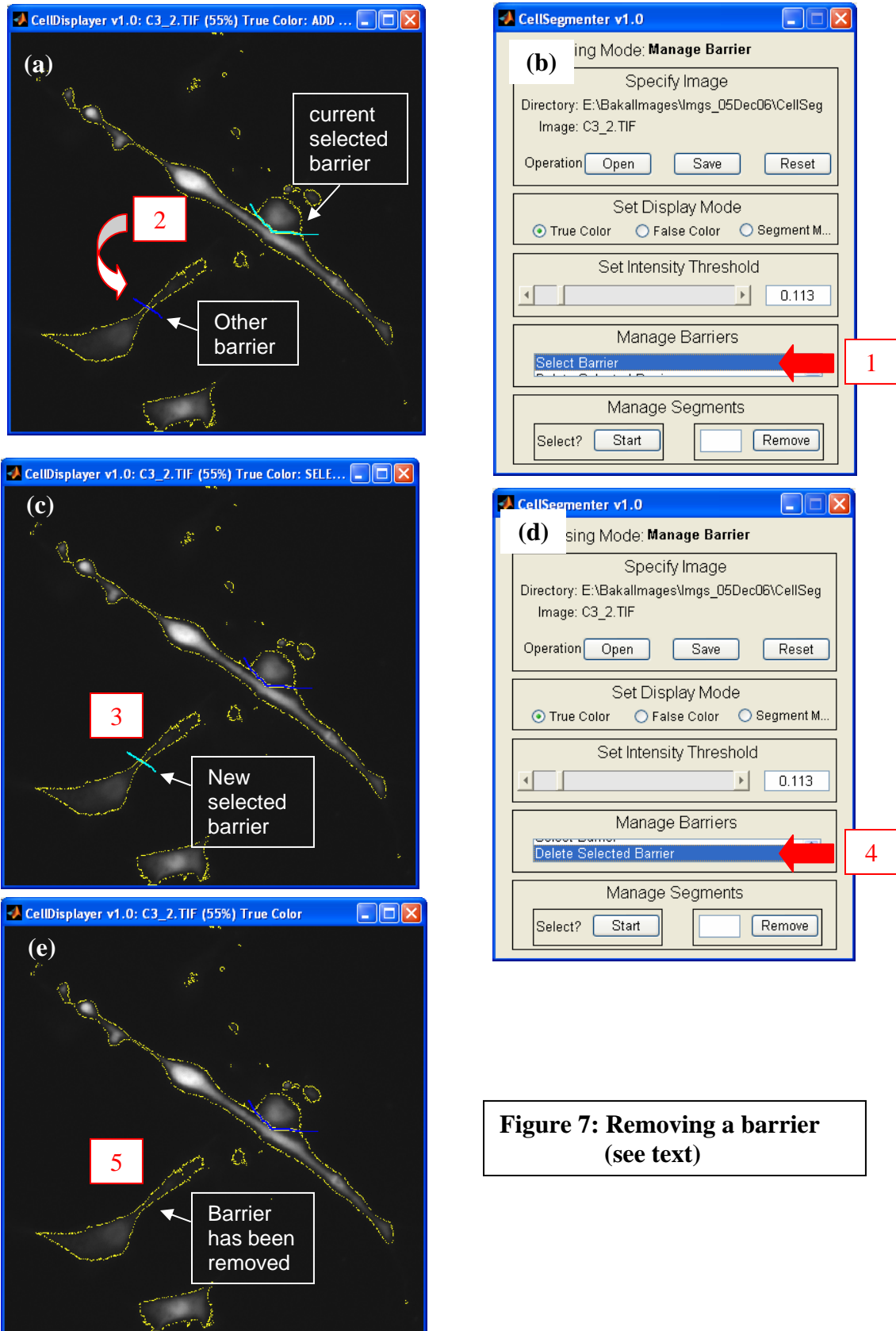
Adding a barrier will have no effect on previously created cell segments. I.e., if one generates a segment for a cell and then draws a barrier across it, the barrier will not be incorporated into the previously defined segment boundary.

The basic procedures involved in removing a barrier are illustrated in Figure 7. The basic operations required are, first, to *select* the barrier to be removed (if required), and, then, to *delete* the selected barrier. Selection of a barrier may be needed because multiple barriers may be drawn into an image over the course of a CellSegmenter session, and it is necessary to identify the one to be removed. The *selected* barrier may be recognized by its rendering as a thick white barrier line, while all other barriers are rendered as thick blue lines (Figure 7, upper left). As barriers are drawn into an image using Add Barrier, they automatically become the selected barrier and rendered as white lines (as in Figure 6), while other barriers become deselected and are rendered with blue lines.

The situation shown in panel (a) of Figure 7 is an image with two barriers, one of which has been selected, but where the barrier we wish to delete is *not* the selected one. The first step is therefore to select the barrier to be deleted. The action needed to do this is to choose the Select Barrier list item in the Barrier Management Area of the CellSegmenter window ([1] in Figure 7b). With this action, CellDisplayer will interpret the next left-button mouse click in the CellDisplayer window as an order to select the closest nearby barrier. This mouse click is indicated as [2] in Figure 7a. CellDisplayer acknowledges this action by rendering this barrier as a thick white line ([3] in Figure 7c), and changing the color of the other barrier to blue (Figure 7c). Now, back in CellSegmenter, the user chooses the Delete Selected Barrier list item in the Barrier Management Area ([4] in Figure 7d). CellDisplayer now processes this action by removing the selected barrier ([5] in Figure 7e).

### **Saving work done by CellSegmenter**

To use the segments created for an image by CellSegmenter in subsequent image processing analysis, they must be *saved* from within the CellSegmenter session for that image. Directing CellSegmenter to save the image causes it to write all information



maintained on the current set of segments and barriers generated for the image to a set of

external files. This information may be saved at any point during a session for an image by clicking on the Save button in the Image Management Area in the CellSegmenter window (Figure 1).

Saving an image results in CellSegmenter writing a ".csf" file for the image and one ".csm" file for every segment. The files are written to the current directory in the Image Management Area, and the prefix of the image (the part up to the .TIF) is used as the prefix of the filenames. For the csf file, the full name is simply this prefix + ".csf". For the csm files, the full names have the form prefix + ".1.csm" for the first cell segment, prefix + ".2.csm" for the second segment, etc.

The csm files are really binary TIF files that represent the image segments formed for each cell segment, as confined to the MatLab "BoundingBox" for these image segments (see MatLab Image Toolbox documentation for information on BoundingBox). The csf file, meanwhile, contains information about the location and size of each of these BoundingBoxes in the original image, and the intensity thresholds that were used to define them. The csf file also contains a series of (row,column) coordinates for each barrier in the image (as barriers are represented as a series of contiguous line segments), as well as an initial line of status information that records the name of the image, the date and time of the save, and other information. The information in the csf file is written in a format that allows it to be read into MatLab and evaluated using the MatLab "eval" command, enabling offline use of MatLab tools in the Image Processing Toolbox for subsequent image analysis.

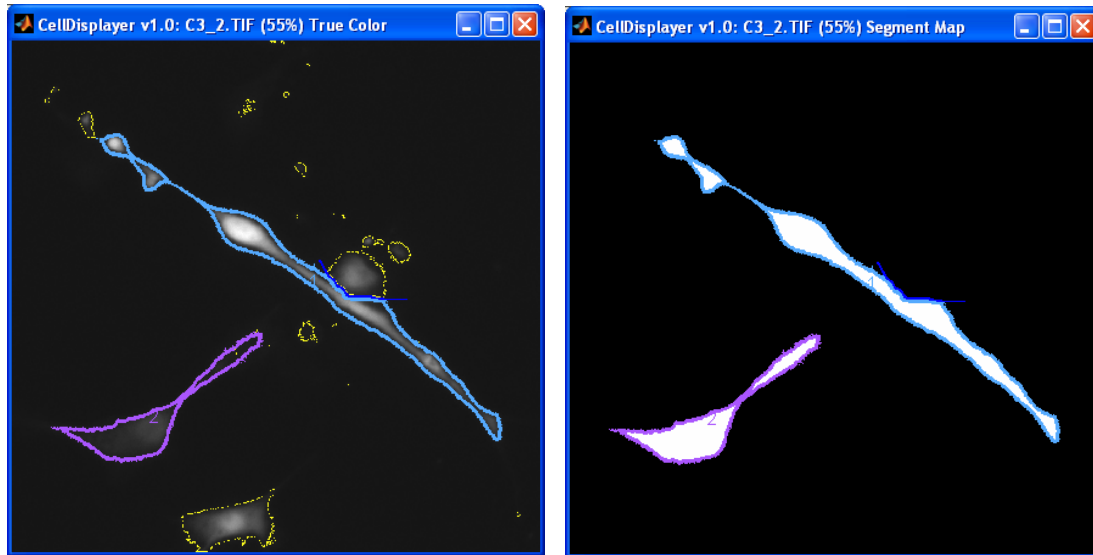
An image may be saved multiple times during a CellSegmenter session. Each save overwrites all previously saved information.

If saved CellSegmenter information is available when an image is opened using the Open button, CellSegmenter reads and reprocesses this information so as to recover the state of the image when the last *save* operation was performed. This allows the user to build up segment and barrier information over the course of multiple CellSegmenter sessions with the image.

## **Other CellSegmenter capabilities**

*Reset:* To abandon work done on an image and start with a fresh, unprocessed copy, click on the Reset button in the Image Management Area. A pop-up window will prompt for confirmation that previously accumulated information is to be deleted. Note that a Reset operation does not recover to the last saved CellSegmenter state, but starts from an entirely fresh, unprocessed image.

*Segment Map:* In addition to the True Color and False Color options in the Image Processing Area, there is also an option to display a Segment Map. This option shows the cell segments and barriers defined for the image only without overlaying these on the original image and current threshold-defined borders. This option can be useful for reading the segment number associated with each defined cell segment, as these numbers can be obscured in True or False Color mode by details in the original image.



**Figure 8: True Color (left) and Segment Map (right) displays of an image.**

A comparison between a True Color and Segment Map of an image is shown in Figure 8.

### **Closing CellDisplayer and shutting down CellSegmenter**

If after working on an image in CellSegmenter, one wants to close the session for this image without saving it, one may simply close the CellDisplayer window in the normal way by clicking on the close window ("X") control in the upper right corner of the window. If there is unsaved information, CellSegmenter will prompt for confirmation that you really want to abandon the image without saving. Closing the CellDisplayer window will reset CellSegmenter to the Initial Processing mode.

To shut down CellSegmenter entirely, first close the CellDisplayer window as just described, and then shut down the CellSegmenter window in the same way. To restart CellSegmenter after shutting it down, you will have to start from the MatLab command line as described above in the "Starting CellSegmenter" section.

### **CellSegmenter .csm and .csf files**

As previously noted, CellSegmenter saves information about selected cell segments in .csf and .csm files when the Save button is clicked. The information in these files is as follows:

#### .csf files

The .csf file contains the locations of and thresholds used to define of all cell segments selected in the cell image, and also the locations of all barriers drawn in the image. The .csf file is written in MatLab statement syntax so that each .csf file line can be read from the file and evaluated in MatLab using the *eval* function. This enables CellSegmenter to reconstruct the segmentation of a previously processed image and so allow users to modify this segmentation. It also allows users to write additional MatLab code to read



and use the .csf, .csm, and original cell TIF image files to generate features for cell segments selected using CellSegmenter for follow-on image analysis.

The first line of the .csf file contains MatLab assignment statements for variables that describe the CellSegmenter program and version that wrote the .csf file, the time the information was Saved, the file name prefix of the image, the numbers of segments and barriers, the number of the currently selected barrier, and the threshold setting of CellSegmenter at the time of the Save.

After the first line, one line appears for each selected cell segment Saved in the image that defines the location of the rectangle containing the segment, the threshold used to define the segment, and the centroid of the segment.

Following all cell segment lines, one line appears for each barrier that was created in the image. Each barrier line is specified as an array of pairs of pixel coordinates in the image that describe the endpoints of the line segments that constitute the barrier. The currently selected barrier, indicated in the first line of the .csf file, is given as the ordinal position of the particular barrier line in the .csf file that corresponds to that barrier.

#### .csm file

The .csm file is simply a binary TIF file that specifies the boundary of a selected cell segment, where this file is stored with the suffix .csm instead of .TIF. .csm files correspond one-to-one with segment lines in the .csf file, so that the first segment has a suffix ".1.csm", the second segment has a suffix ".2.csm", and so forth, all with the image file prefix specified in the first line of the .csf file and the .csf file name itself. The dimensions of the image stored in a .csm file are identical to the rectangle dimensions specified in the segment line in the .csf file that corresponds to the .csm -- with the .csf file providing the additional information needed to locate this rectangle in the original TIF cell image.

## **Citing CellSegmenter**

When using CellSegmenter, please cite our manuscript

Bakal, Aach, Church, Perrimon (2007) Local Signaling Networks That Regulate Cell Morphology Defined by Quantitative Morphological Signatures, *Science*

## **Copyright**

Copyright © 2007 by John Aach and the President and Fellows of Harvard University

CellSegmenter is available for download and use under the terms specified in the program download page linked to <http://arep.med.harvard.edu/QMS/>.