

GUIDE TO LCR ANALYSIS PROGRAM – XYTEventDetector

1. About the Program

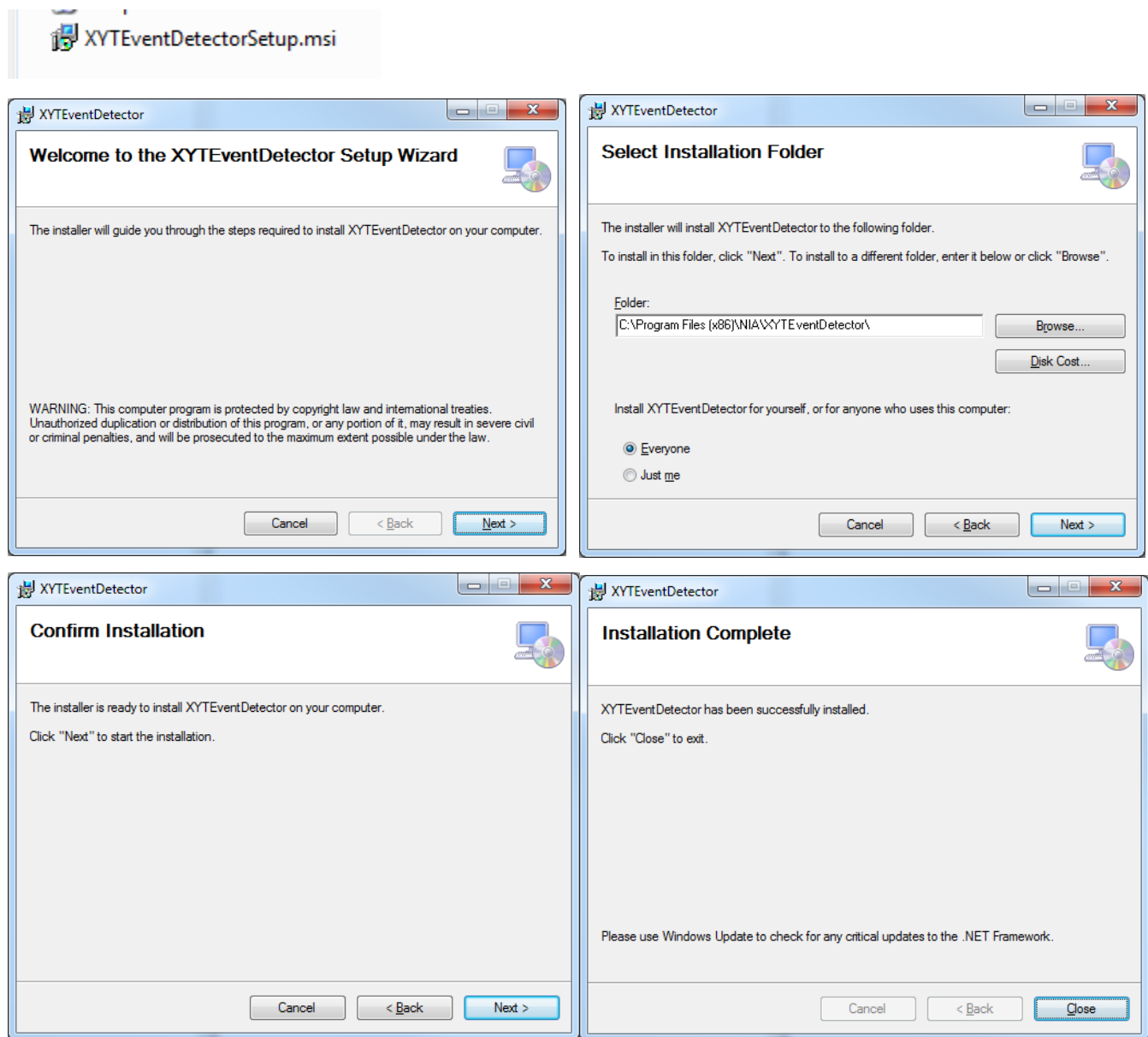
The program was created by Alexander Maltsev as a part of his work at the Laboratory of Cardiovascular Science at IRP/NIA/NIH in Baltimore, USA in 2015-2016. The program was tuned to work with real cells by several researchers, Oliver Monfredi, Mary S. Kim, and Kenta Tsutsui. The program interface and input/output parameters are linked to SA node cells and specific recording conditions, including Ca indicator Fluo4, 63x oil-emersion lens, and high-speed CCD camera (Hamamatsu C9100-12). Thus, the program is still in its beta stage and its functionality needs further tuning, especially if another cell type, Ca indicator, or high-speed camera is used.

Since the program was created at the NIH, i.e. a US Government agency, it belongs to a public domain and can be freely distributed for the purpose of improving healthcare in the US and worldwide. In this research paper, we provide the entire original program code and all libraries (OpenGL, tif file handling, etc.) that allow the code compilation within the Visual Studio 2010. The result of the compilation is a msi setup file that can be used to install the program on a computer with a Windows operating system (we used Windows 7). If you use or develop further our software, please give a credit to our work by citing our Research paper.

In short, the program works with stacks of tif images of Ca signals measured in living cells. The consecutive images in the stack represent the intensity of local Ca dynamics measured with a fixed sampling rate by a high-speed camera and a fluorescent Ca indicator. When pacemaker cells generate a cardiac impulse, the action potential induces a global Ca increase in cytoplasmic Ca (dubbed AP-induced Ca transient). The program identifies and eliminates global signal shifts and transients and detects only local signals.

2. Program Installation

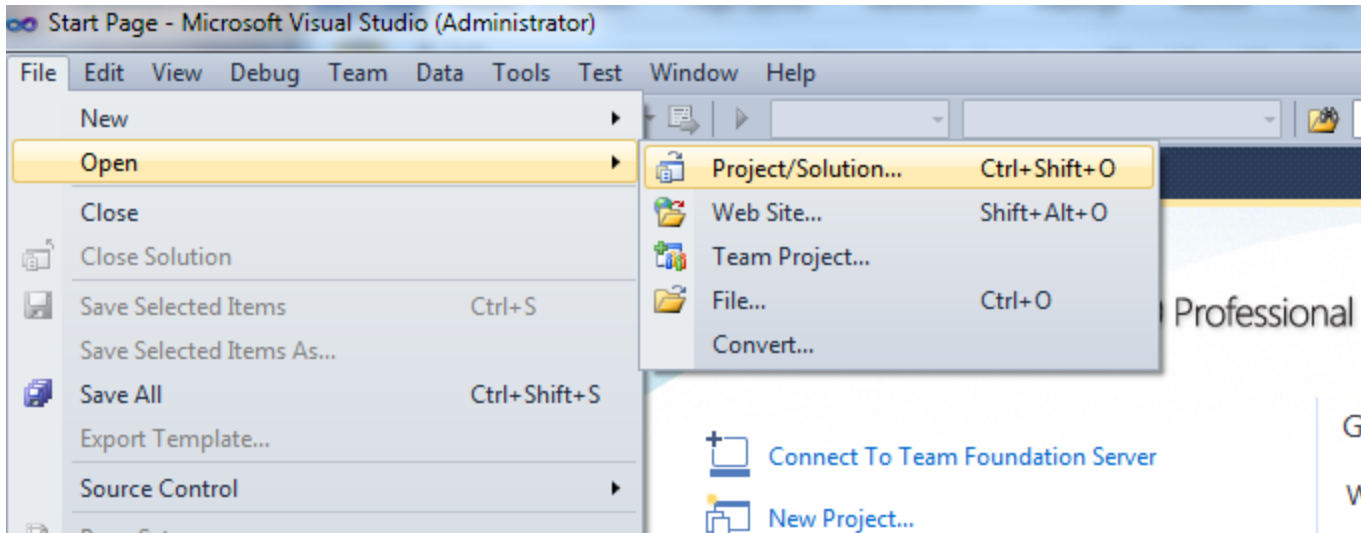
We provide msi file XYTEventDetectorSetup.msi that installs the program fully automatically in computers with Windows operating system. The user should be administrator.



3. Program Compilation of our original C++ code within Visual Studio 2010

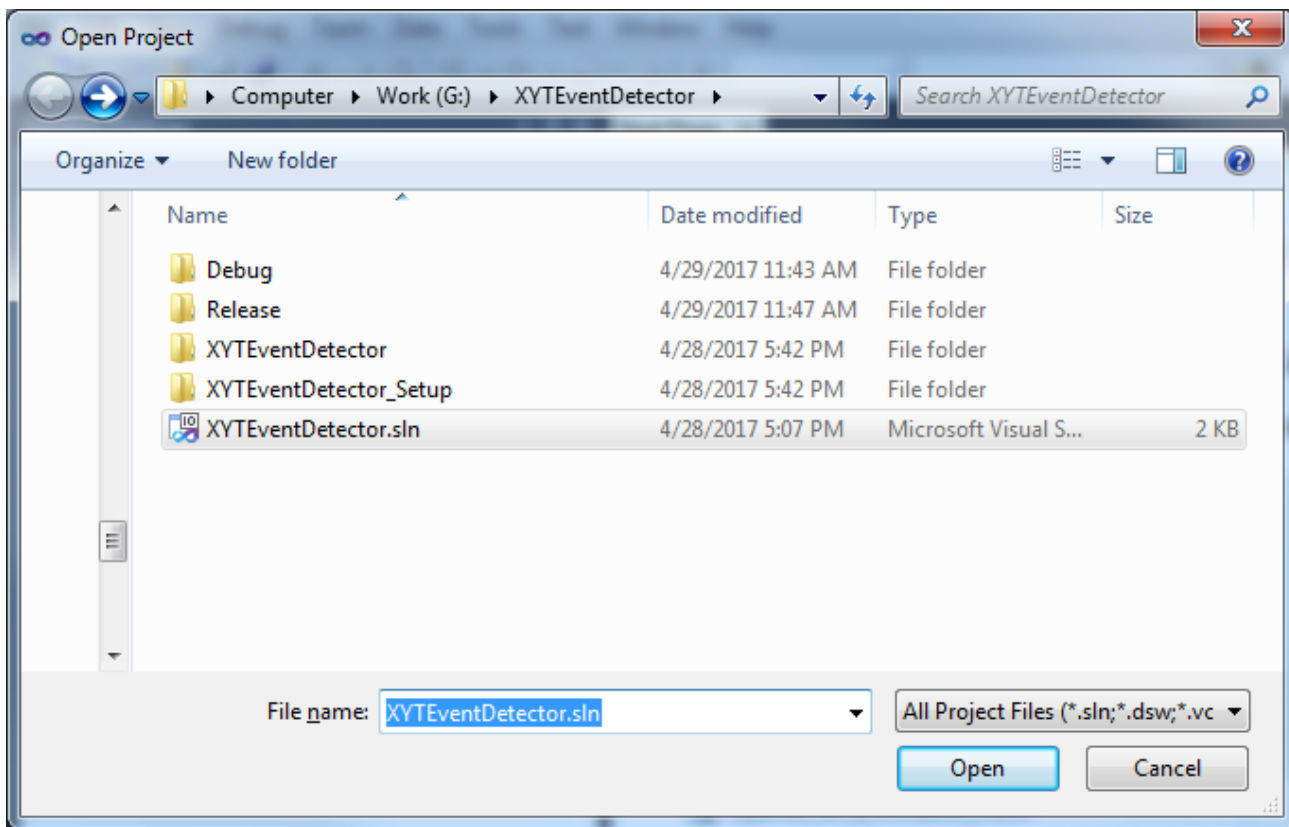
If anyone wants to recompile the program from our original or custom- edited C++code, this can be done in Visual Studio 2010 as follows. We provide the entire directory with our code and libraries in a Zip file with the name having date of a given release, for example XYTEventDetector-2017-04-28.zip.

Unpack this zip file, start Visual Studio and choose in File Menu Open → Project/Solution

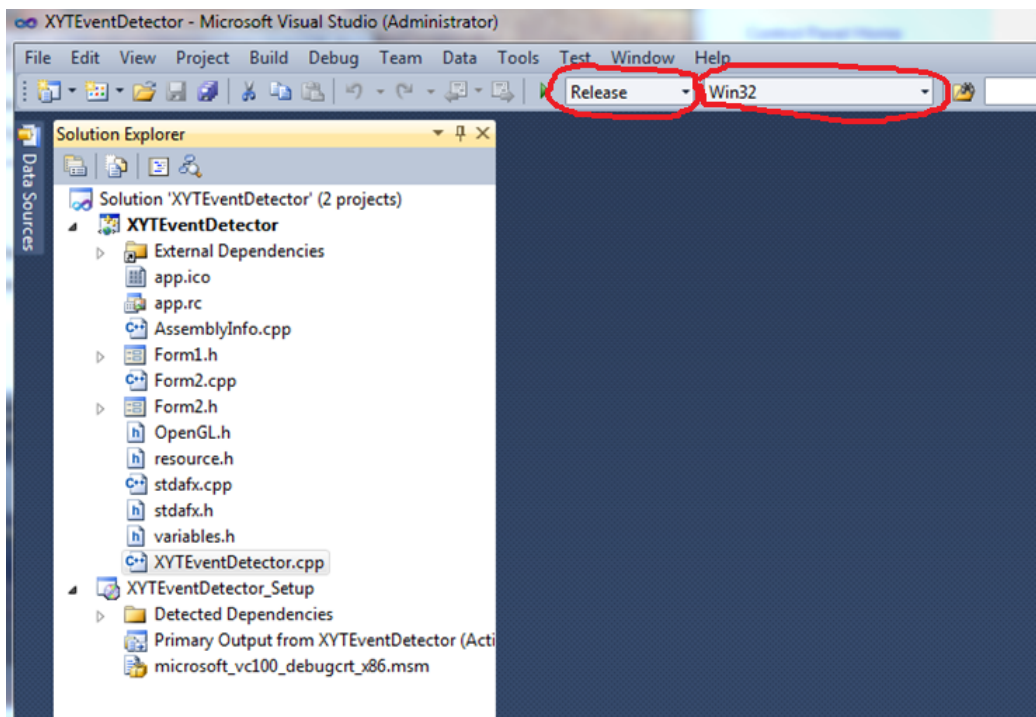


Then choose file XYTEventDetector.sln in unzipped folder structure in directory

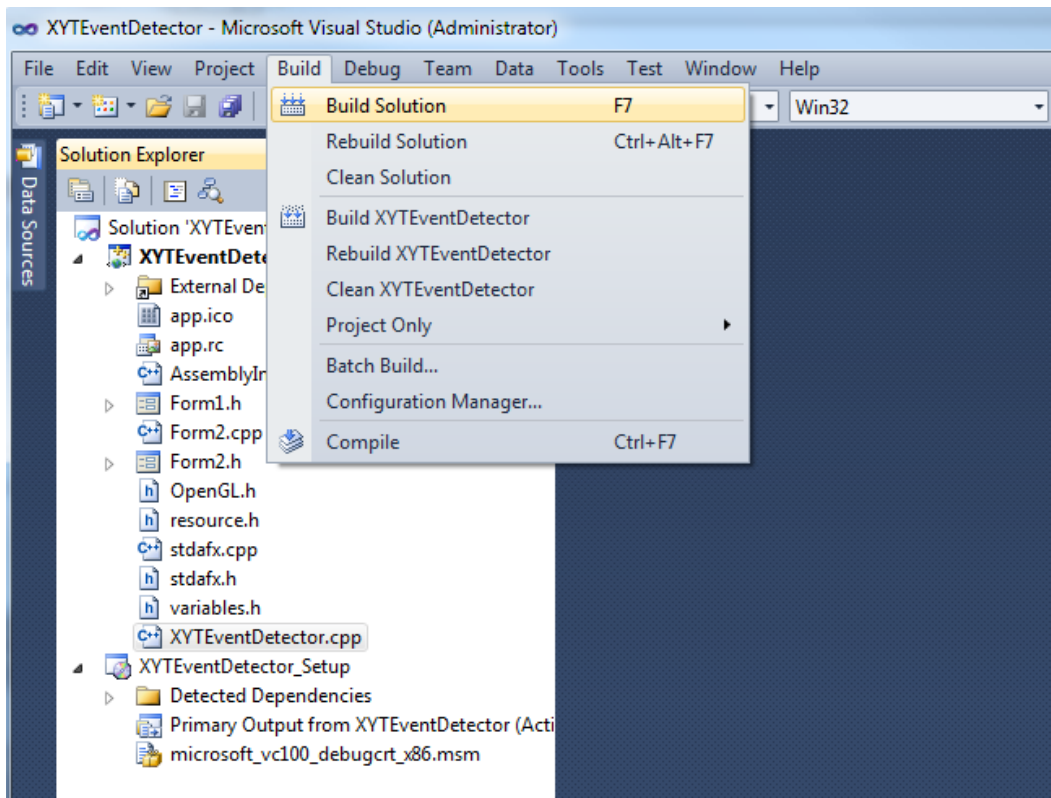
... \XYTEventDetector



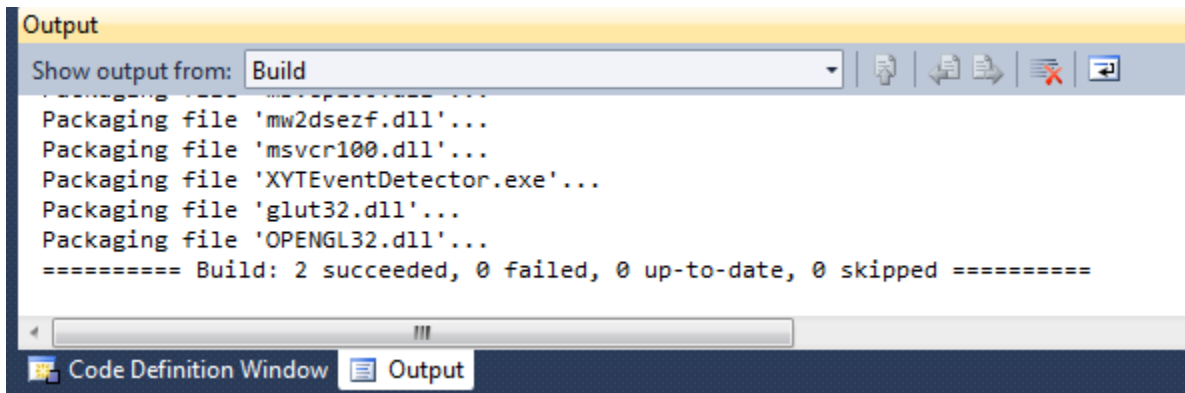
The project will be loaded to Visual Studio. It should be Release and Win32



To compile the project use F7 or Build Solution menu option

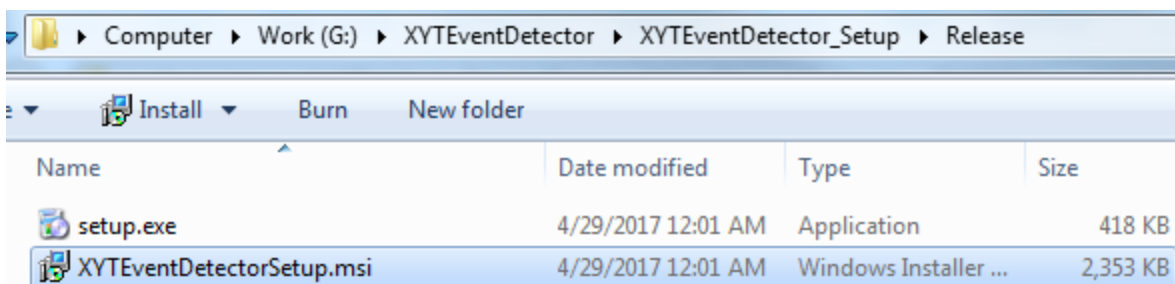


Successful compilation should show message “Build: 2 succeeded,...” at the end.



```
-----
Packaging file 'mw2dsezf.dll'...
Packaging file 'msvcr100.dll'...
Packaging file 'XYTEventDetector.exe'...
Packaging file 'glut32.dll'...
Packaging file 'OPENG32.dll'...
==== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

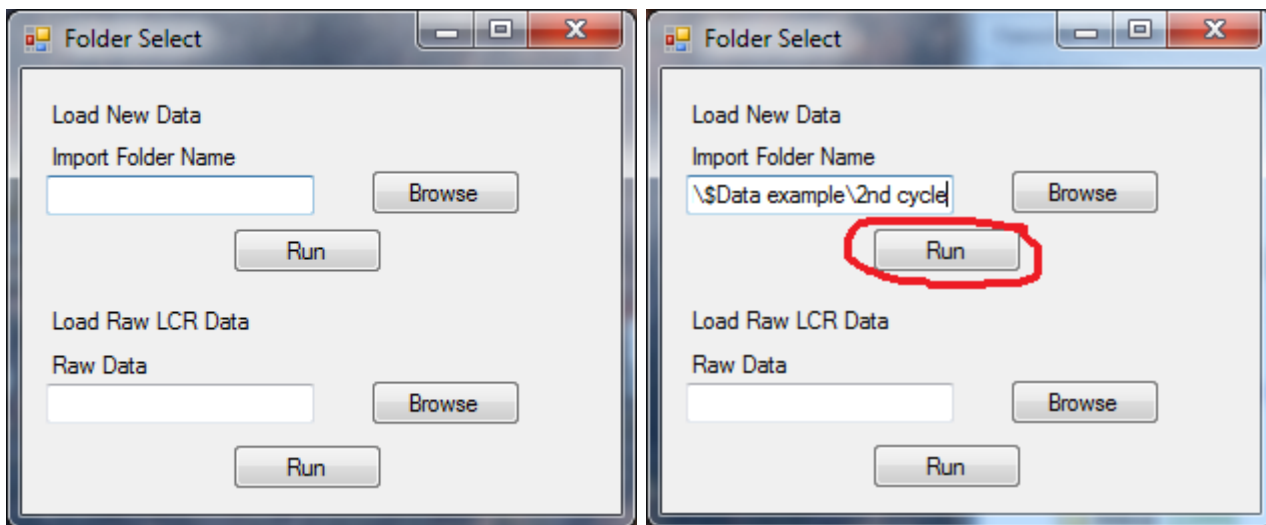
Please then locate freshly compiled file installation file XYTEventDetectorSetup.msi in directory ...\\XYTEventDetector\\XYTEventDetector_Setup\\Release



This file can be used for program installation in Windows computers as described above in section 2

4. Loading New Data

After launching the program XYTEventDetector, a “Folder Select” window will appear. To load new data, copy-paste the folder path into the text box or click the browse button and find the proper directory.



Click the top “Run” button to load the new data. For our purposes the program could load about 2000 8-bit images of size 500x200 pixels. If more frames must be loaded in a recording, find an image with the peak of a transient (since there will be no LCRs there) and continue loading and analyzing images from that point.

5. Loading Saved Data

At the first window that pops up “Folder Select,” input the original files that you were working with (the TIF folder) in the first box just as above and then the raw saved file (.txt) in the second box below. Make sure the raw saved file matches the correct TIF folder. Then click the bottom “Run” button. Do not filter data after loading saved LCRs. If you attempt to use the **Find LCRs** button, it will only find the LCRs that were originally imported (it works sort of like a reset button).

6. Working with New Data

After loading a new window starts with the program interface and cell image (see example below). This data example is provided in a zip file data_example_01.zip with the program package for testing and illustration purposes. “Transients” checkbox informs program that data has transients. If this box is unchecked programs analyses LCR activity regardless to pacemaker cycle, e.g. in permeabilized cells.

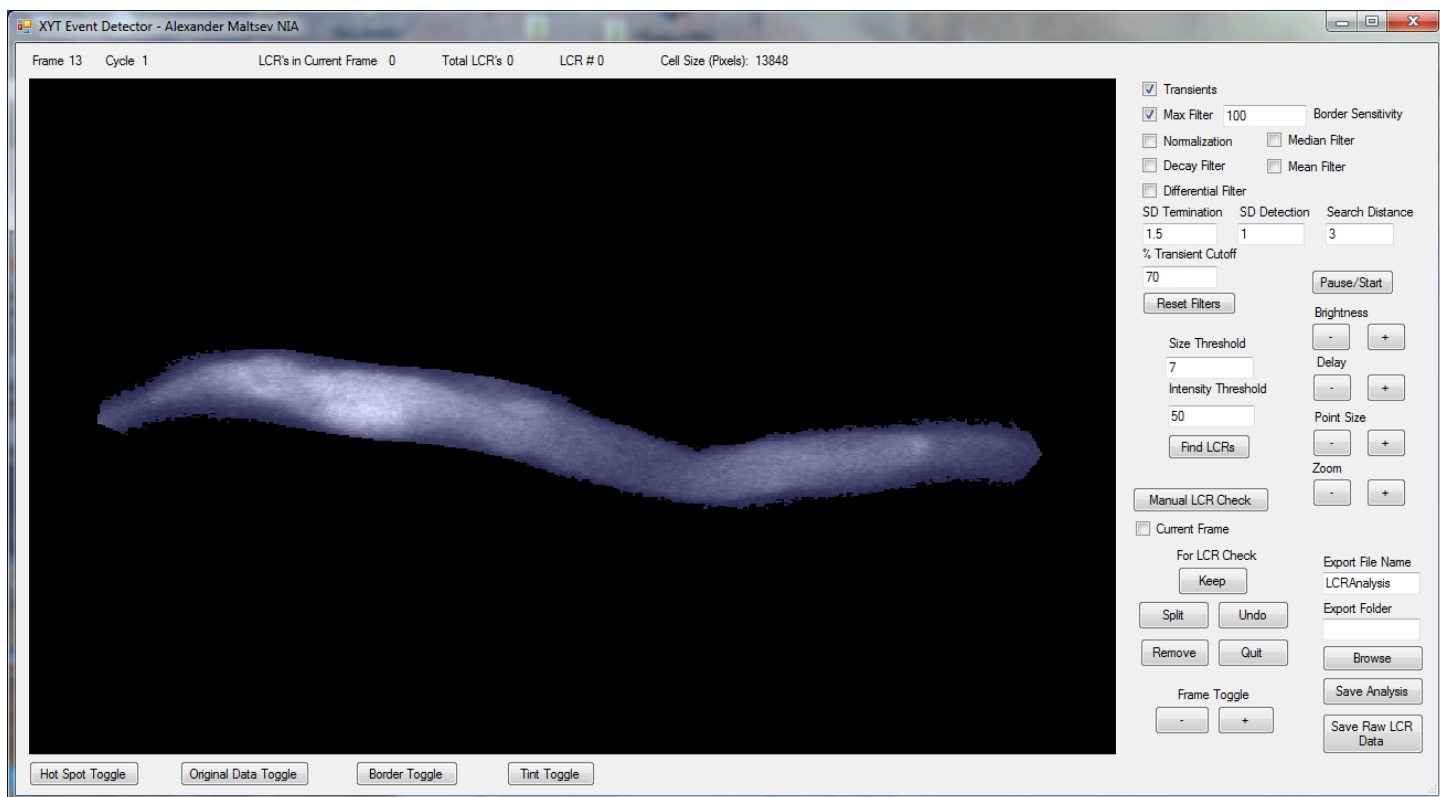


7. Filtering

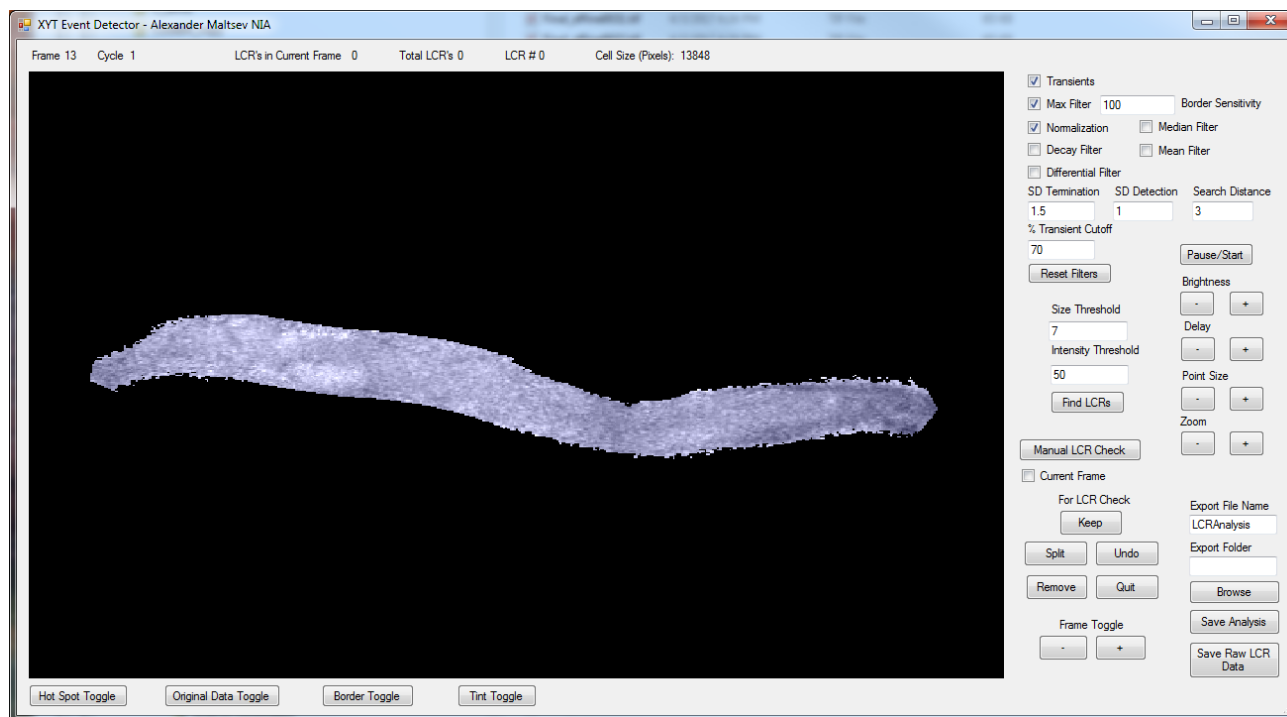
The filters are engaged when respective box is clicked. The sequence of filter application is from top to bottom as they appear in the interface. The program does not modify the original loaded files on the hard drive when filtering because it copies the files into memory and works with the copied data. The program has an option to reset filters but it needs two clicks to work. Another obvious option is to close the program and reload the data, which also works really quickly.

Important note: if the program seems to be frozen, especially executing “Differential Filter”, it is not broken and most likely calculating something. Do not try to click buttons to fix it, because more workload will be added to the program and will cause more problems.

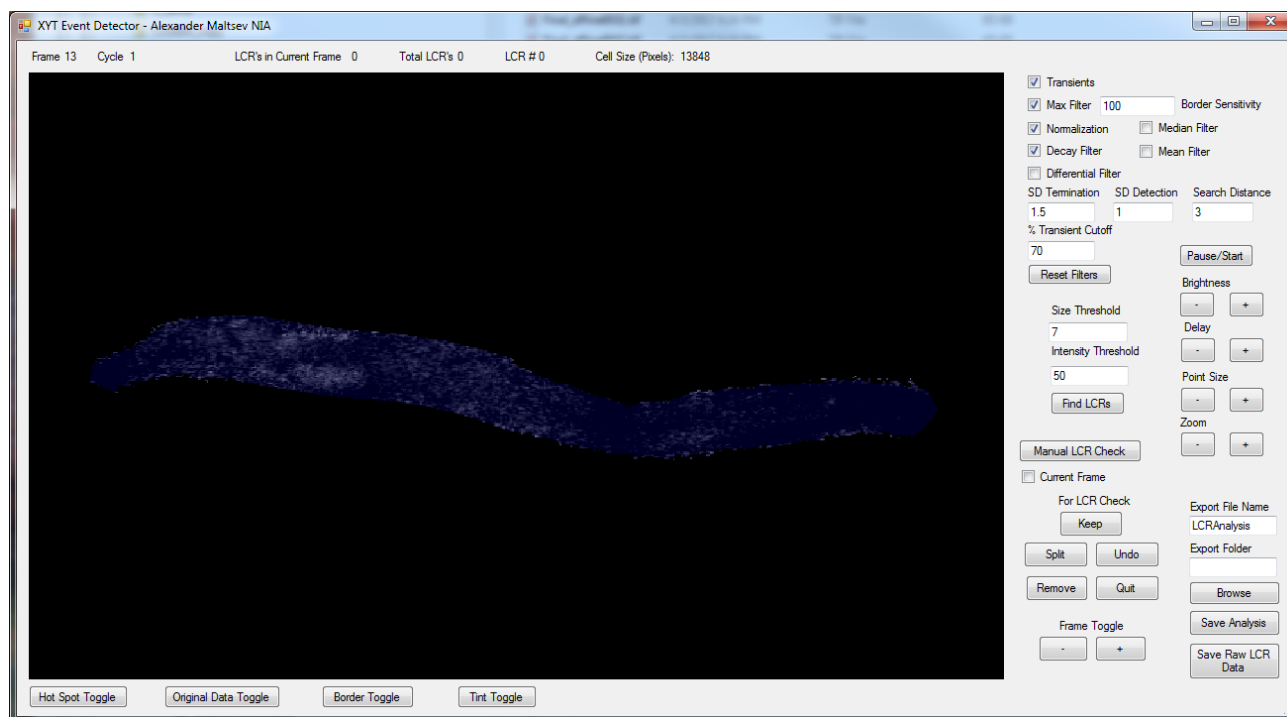
7.1. Max Filter - isolates the cell area. It calculates the maximum value of each pixel over the entire stack of images and then takes the mean of the all the maximum values. Whatever is lower than this mean is considered part of the background noise and whatever is higher is considered the cell. This filter will also tint the cell to show where the cell is located. LCRs will not be found where there is no tint. There is a box next to Max Filter which specifies the percentage of the mean that should be taken as the cut point. Decreasing this number will expand the cell and increasing it will shrink the cell (the tint should be used as a visual indicator).



7.2 Normalization - calculates the minimum and maximum for each pixel time series over the entire stack of images. Based on those values it then scales all the pixel values to a range of 0 – 255.

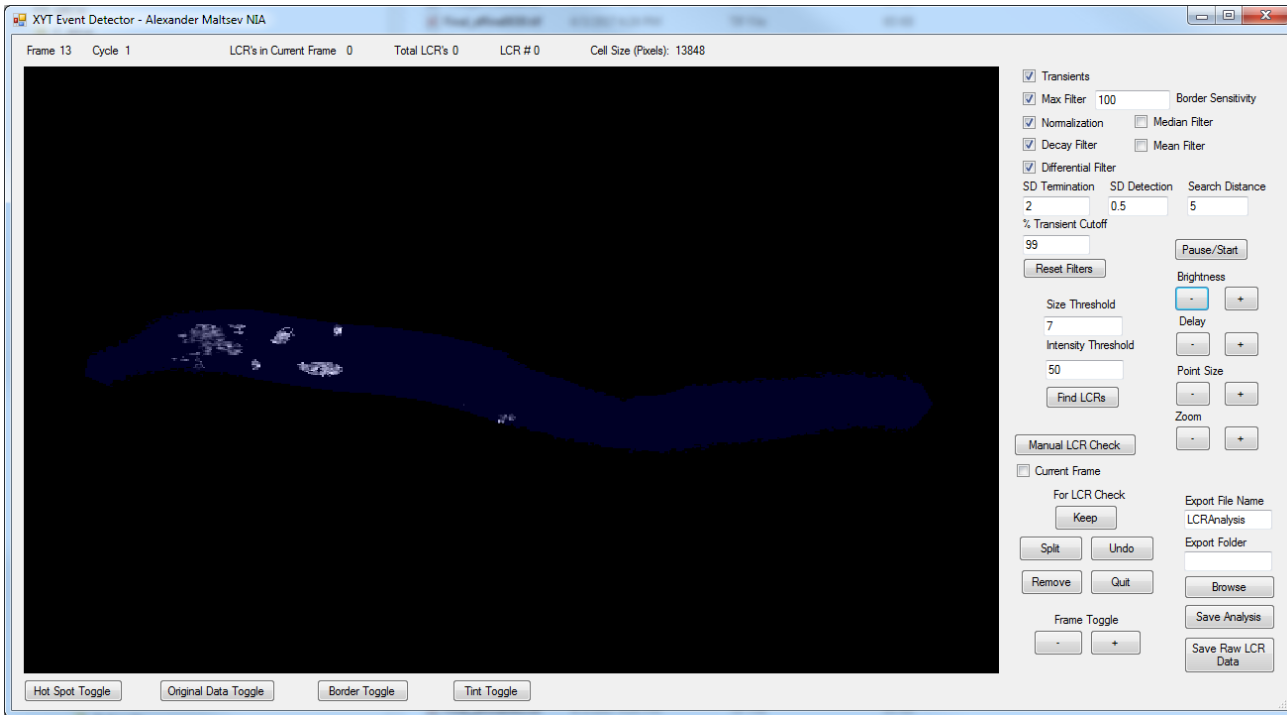


7.3 Decay Filter. If there are action potential-induced Ca transients in the data, then the global signal of the cell will be either decreasing or increasing. This function gets rid of the global signal fluctuations.



7.4 Differential Filter. It is important to note that this filter is a key part of LCR detection that is tuned here by LCR detection sensitivity parameters “SD Termination”, “SD Detection”, “Search Distance” and “% Transient Cutoff”. Therefore please set their values in the interface before its use.

This filter goes through all the pixels and compares the summed intensity at an area around each pixel at frame f to the same area around the pixel at frame $f-1$. It is very helpful in detecting LCRs, because whether an LCR is detected or not depends on how much the signal in an area increases in two consecutive frames that is actually this filter evaluates. After the spatiotemporal filtering, this filter performs further important discriminations based on signal strength in finding LCR, LCR progression, and LCR termination. This function can be customized by tuning parameters described below for specific recordings with different signal strength over noise due to various indicator loading, excitation light intensity, cell type, different quality of optics, cameras, etc.



7.4.1 Search Distance

Differential filter works first as a spatiotemporal filter that integrates signals spatially with the search area defined by Search Distance (default is 3) around each pixel and then differentiate in time, as it calculates frame-to-frame difference from these spatially averaged frames. The search area s is therefore linked to Search Distance as $s = (\text{Search Distance} + 1)^2 = 7 \times 7 = 49$ (by default setting of Search Distance). Decreasing this

number leads to detection of more LCRs. We would normally start with Search Distance set at 3, but it can range between 2.5-6 in our experience.

7.4.2. SD Detection. After spatiotemporal filtering, this filter determines whether the pixel intensity is strong enough compared to Standard Deviation to be further considered as a part of LCR. Its default value is set at 1.0. Decreasing this number also leads to detection of more LCRs as well as noise; 0.4 is a good starting value, though the range of values that we have found useful in different recording circumstances has been 0.1-0.8.

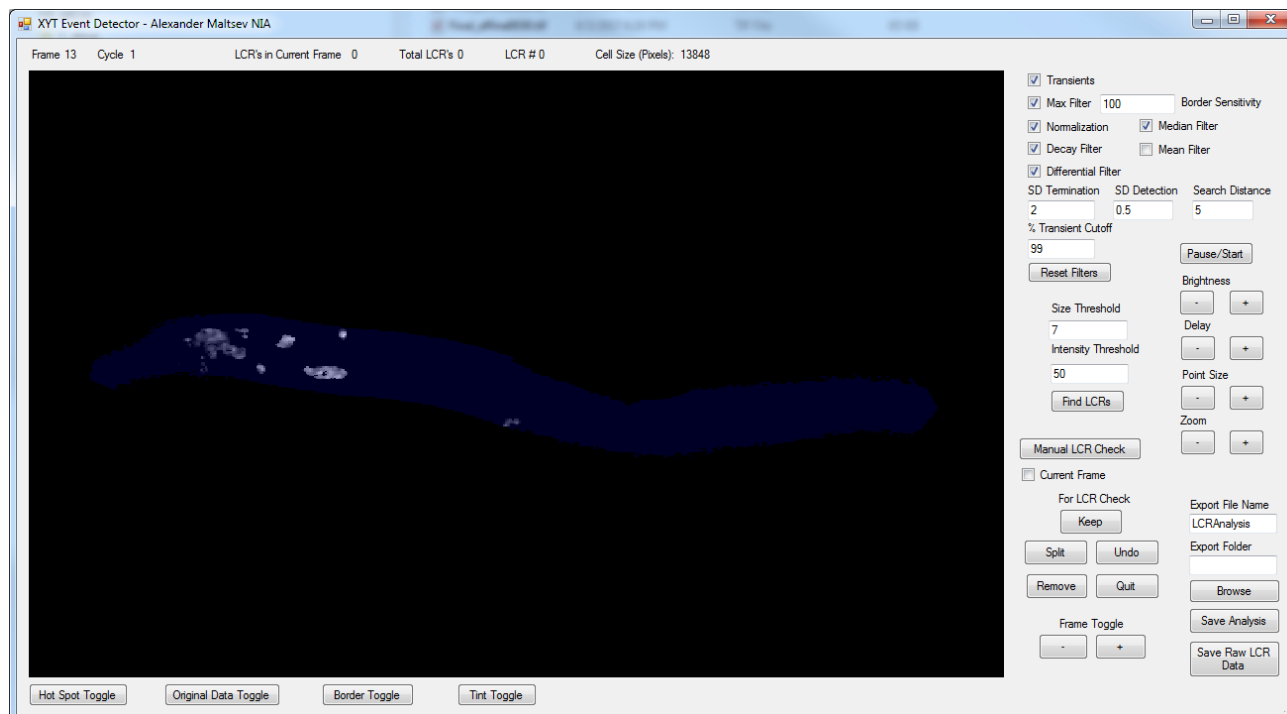
7.4.3 SD Termination determines whether the pixel intensity drops low enough to no longer be part of an LCR. Set at default at 1.5, if the sum of the intensity in the area at frame f is 1.5 standard deviations (of the cell signal) less than the first sum of signal (in the LCR), then the pixel is no longer part of an LCR. SD Termination affects LCR duration; increasing from the default terminates LCR detection sooner and can be useful if one finds that LCRs are being detected after they have overtly terminated. In practice, we have found that this rarely needs to be altered.

7.4.4. Extending LCR candidacy by neighborhood. We also assume that if a pixel is inside a possible LCR then it is also part of the LCR. Thus, to account for pixel anomalies inside LCRs, we have assign LCR candidacy to a pixel if 7 neighbors around it have LCR candidacy. This function works behind the scenes and does not have a customizable parameter in interface.

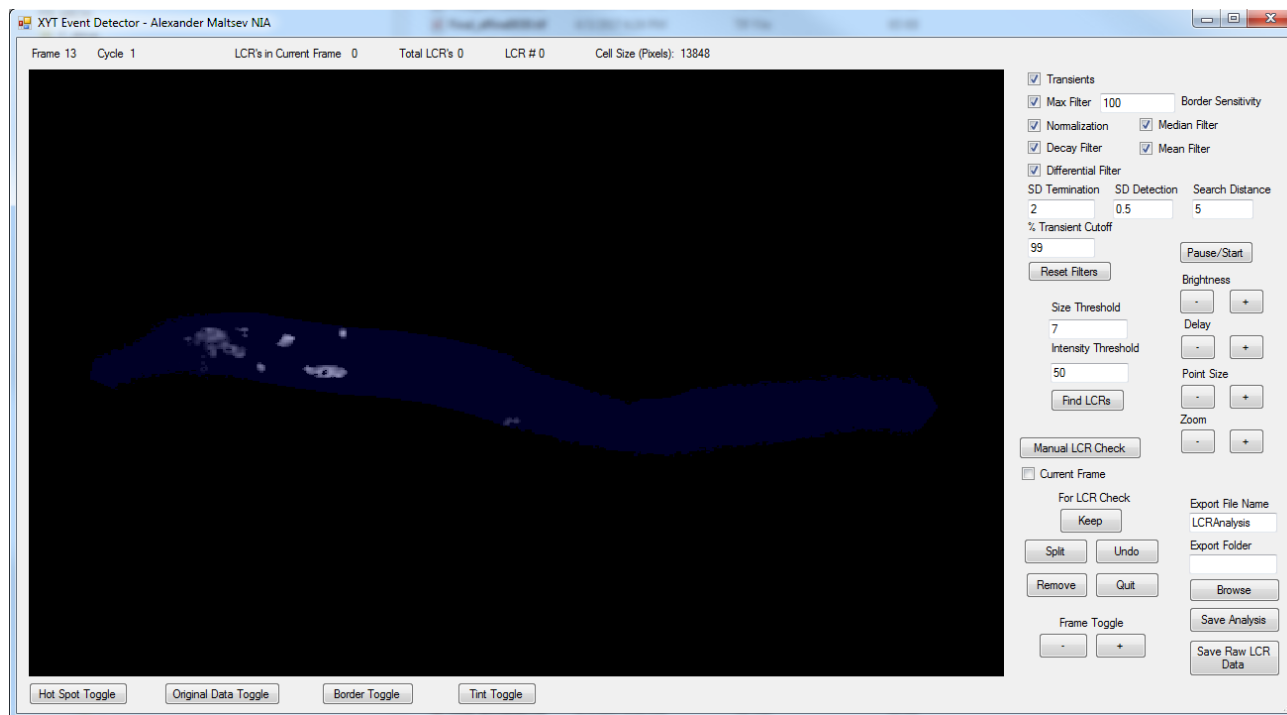
7.4.5. % Transient Cutoff. There will be LCRs that grow into the transient and thus it is important to cut them off before they become part of the transient. When the LCR grows to the inputted percentage of the cell size, the LCR is considered terminated by the transient. Thus “% Transient Cutoff” defines the level of onrushing transient at which the program stops looking for LCRs, and accepts that the Ca^{2+} transient has overwhelmed and engulfed the cytoplasm to the extent that LCRs can no longer be detected. We usually use 98 or 99. If the cell does not have transients, the box “Transients” should be unchecked and it does not matter what is typed into the “% Transient Cutoff” box.

There are 2 additional (optional) filters: a median filter and mean filter to get rid of any remaining noise. This filter produce rather minor effect as their width is limited from a given pixel to nearest neighboring pixels.

7.5. Median Filter. This filter should be used if there still exist some noise or in the event if there exist LCRs that are connected by small strings of pixels which could not be removed with any other filtration. This option ends up cutting borders of LCRs and should be used only if absolutely necessary. It finds and sorts all the pixels 1 above, below, right, and left of the checked pixel and then sets the checked pixel to the middle value of the set. Here is the effect of median filter:



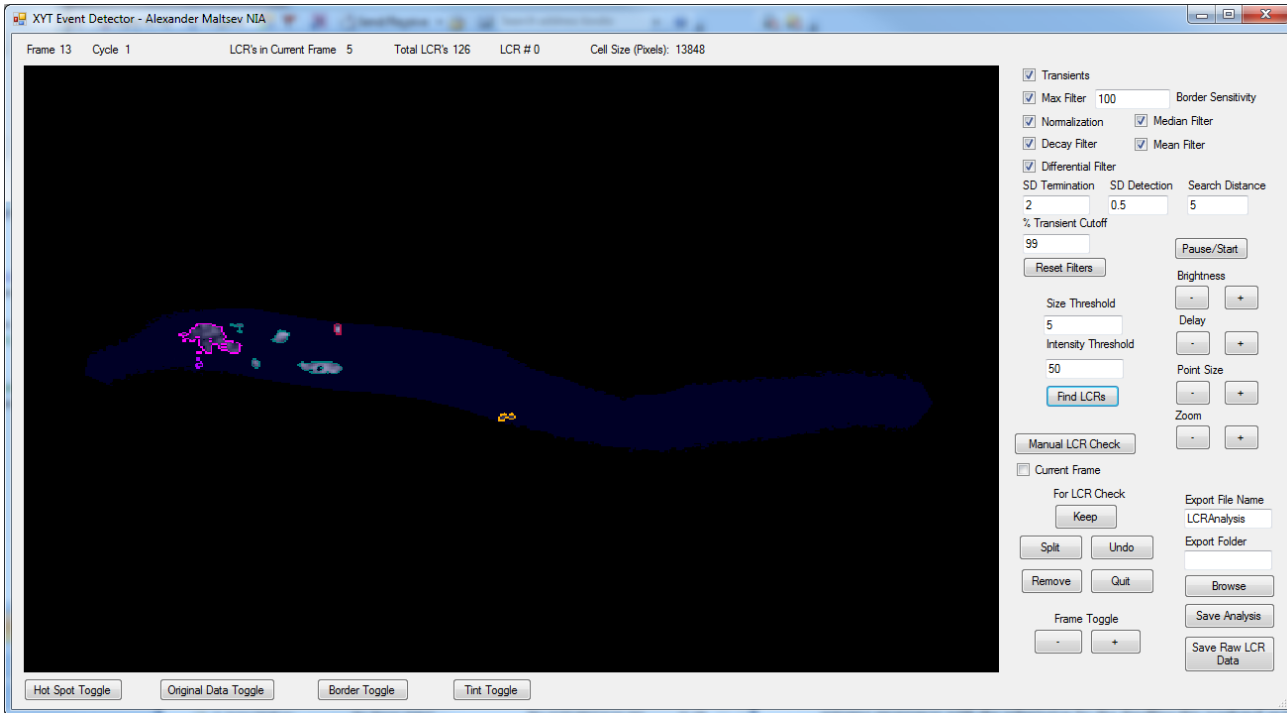
7.6 Mean Filter. This filter's function is primarily used for aesthetic to improve visibility of peaks in LCRs because it only targets pixels that have fluorescence. The values of the pixels 1 above, below, right, and left of the checked pixel are examined (if they have fluorescence) and then the value checked pixel is changed to the mean of the set. And this is after additional mean filter:



8 Finding new LCRs.

After all filtrations discussed above, any remaining signal is classified as possible LCRs.

Final LCR detection starts with pushing “Find LCRs” button in the interface. In addition to LCR detection sensitivity parameters within Differential Filter described above, this function has 2 additional sensitivity parameters to further tune LCR detection for best performance to reject noisy smaller LCRs or for a specific aim of a study (for example, to study selectively only large LCRs). The program searches for clusters, i.e. sets of pixels having common borders, which would satisfy specific conditions described below.



8.1. Size Threshold A newborn LCR, a cluster of pixels of any morphology, at a certain frame must have a minimum number of pixels defined by this parameter. In practice, this is what the operator considers to be the minimum starting size of an LCR in pixels. Increasing the value gets rid of noise but risks failing to detect smaller LCRs.

8.2. Intensity Threshold. A newborn LCR must be sufficiently bright on a scale of arbitrary units 0-255. If LCRs are dim, this number should be decreased, and vice versa. In practice, we rarely change this from the default as it has few if any consequences to the results.

It is important to note that the two sensitivity tests defined by **Size Threshold** and **Intensity Threshold** act together with Boolean function AND, i.e. a newborn LCR must be large and bright enough to pass these tests otherwise it is considered as noise and discarded.

9. LCR progression and termination

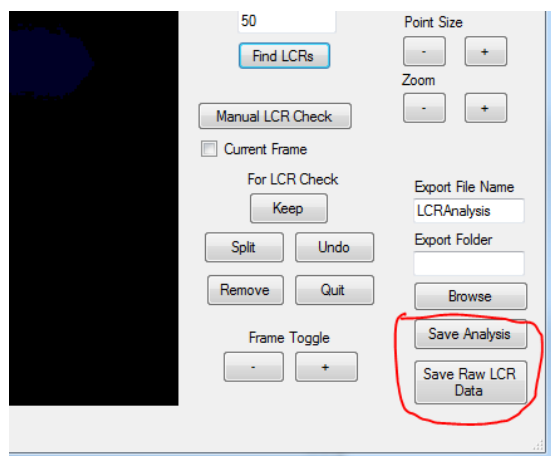
After a pixel is considered a part of an LCR, it will continue to be part of that LCR until the LCR is presumed terminated. In the case of LCR decay, a pixel is considered part of a possible LCR until an average signal in its series becomes less than a factor of the standard deviation that is also by default set at one in the interface parameter “SD Termination” that used by Differential filter, described above.

10. Hints how to use input parameters

If cell still contains a lot of noise after using the differential filter then redo the filtration with a higher **Search Distance**. If it seems like there is not enough signal in the cell, decrease the **Search Distance**. If LCRs seem very small, try slightly increasing the **SD Detection**. However if LCRs seem too big, try slightly decreasing the **SD Detection**. There may also be a problem where an LCR die too slowly or may seem to never die. You could then try to increase the **SD Termination**. There is also a different way to handle the problem using **Split** and **Remove** in **Manual LCR Check**. If LCRs seem to die too quickly, decrease the **SD Termination**.

11. Saving Results and Raw LCR Data

MAKE SURE TO SAVE THE LCR ANALYSIS AND THE RAW DATA FILE UNDER DIFFERENT NAMES OR ELSE FILES WILL BE OVERWRITTEN AND LOST.



Save Analysis – Saves the results of the LCR analysis into the **Export Address**.

Save Raw LCR Data – After finishing analysis of a file, make sure to save the raw pixel data of each and all LCRs. This will give you a raw file of all the data needed to restore the LCRs in the future. Input a name for the raw save file under **Export Name** and a destination for the file under **Export Address**.

12. Modifying the Display

Pause/Start. It pauses or starts the movie with LCR (when and if found)

Frame Toggle – Allows you to manually flip through frames (when paused)

Brightness – Increases/decreases the brightness of the entire set of images. This could be used to see if there is any very small noise that is near dark and unnoticeable. Another possible use for this feature is if two movies seem to vary greatly in brightness and you want to be able to compare them.

Delay – Change the speed of the movie

Point Size – Occasionally if a large image is imported, the image may appear gridded and pixels are not connected. Increasing this will connect the points to get rid of the grid.

Zoom - Zooms in and out of the image. If the image begins to appear gridded like described above, increase the **Point Size**.

13. Manual Checker

The program allows manual check of the detected LCR activity. Check box “Current Frame” allows to check the LCRs only in a certain frame.

Keep – Keep the LCRs currently outlined

Remove – Remove the LCR currently outlined.

Undo – Undo the remove that you just did.

Split – Split the current LCR outlined. You will remain on the part of the LCR that is before the split time.

Quit – Quit the Manual LCR Checker. REMEMBER TO QUIT THE MANUAL LCR CHECKER BEFORE USING A DIFFERENT FEATURE.

The **Manual LCR Checker** can be useful in some specific situations described below.

- A new LCR may spawn where another LCR is dying and thus it looks like the dying LCR spontaneously began to regrow. This case can usually be clearly distinguished from natural LCR growth and can be handled as follows. The user should go to the time when the apparent new LCR spawns and **split** the event. The program now records the previous dying LCR as dead at that frame and the new LCR as starting at the frame after.
- Sometimes increasing **SD Termination** still may not end up terminating an LCR and it continues to live forever. The user can **split** where the LCR becomes clearly dead and **delete** the part after the split. Or if the part that is split off follows the case described above where it grows into an actual LCR much later, it should be **split** again and the part before the split should be **removed**.

14. Toggles

Hot Spot Toggle – Turns on/off hotspots. This is a feature that our team is exploring which displays where LCR activity is most centered around.

Original Data Toggle – The original data is displayed rather than the filtrated data.

Border Toggle – If the user is in **Manual LCR Check**, you can turn off/on the border of the LCR being looked at to have a better look at it. Its highly recommend to also have the original data on (using the toggle above) to have a better look.

Tint Toggle – Turns on/off the blue tint of the cell area.