KESM Suite

Software Suite for KESM

This document is specifications for KESM Suite. Written on 10/12/2011.

# Terminology

## Specimen

Specimen refers to a tissue embedded in plastic. The size is limited to 15mm in length (x-axis), 12mm in width (y-axis), and 9mm in height (z-axis).

## Plank, Plank width, Plank Depth

Plank is a stack of images cut as one stair step. Plank Width is the width of an image. Plank Depth is computed from the number of consecutive sections.

[Add Stair-step cutting image]

## (Tissue) Ribbon

## Image Stack, Column

Image Stack is a stack of images from one y-axis location. Image Stack is interchangeable with Column.

# Coordinate System

## Stage Coordinate System

The stage conventions were set by Aerotech, Inc.

x-axis stage :

y-axis stage :

z-axis stage :

## Specimen Coordinate System

column-axis :

depth-axis :

cutting-axis : (Left knife edge --> column # starts from your right side.)

# Scanning

The knife does not move. The stage moves so that the specimen goes toward the knife edge.

A specimen is scanned along the positive cutting-axis as the stage moves to the positive x-axis stage. The stage movement is left to right when an operator stands in front of the KESM. When the left side knife edge is used to cut, the column-axis starts from a distant location.

# Data Storage and Indexing

## Data Storage

Each Image Stack is saved in a separate folder named a sequential column number. In each folder images are stored in a special naming convention. This guarantees a unique name so it can be used as an identification of an image.

YYYYMMDD\_HHmmSS\_xNyNzN\_tN\_vN

where YYYY: year, MM: month, DD: day, HH: hour, mm: minute, SS: second, N is any real number.

Data information is managed in XML format.

<?xml version="1.0" encoding="ISO-8859-1"?>

<kesm version="1.0">

<date>

<year>2008</year>

<month>01</month>

<day>23</day>

</date>

<specimen>

<animal>mouse</animal>

<organ>brain</organ>

<description>Whole mouse brain vasculature</description>

</specimen>

<pixel>

<width>0.625</width>

<height>0.7</height>

</pixel>

<cutting>

<edge>right</edge>

<columns>6</columns>

<distance>1500</distance>

<depth>1</depth>

</cutting>

</kesm>

The Ribbon width can be calculated from this information.

W (pixel in image) = Distance between Column / Unit Size of one pixel = 1500u/0.625u = 2400 pixels.

## Data Indexing

Raw images do not have meta information for the data. The tissue area from a raw image must be cropped and indexed.

The number of images in a column is most likely different from in other columns.

To simplify explanation for this, let us assume the tissue on the front face happen to be an image that has maximum tissue height. Scanned images above the dotted line (B) in Column 0 can be discarded since no images there has tissue. Yet we can do this in Column 3 since there are tissues in those images. Also there may have improperly scanned images due to technical difficulties. We cannot simply label serial numbers to each image.

\* Find the upper and lower limit

\* Calculate number of meaningful images in each column.

\* Index 0 starts at the upper limit and increases toward the positive depth-axis.

# Image Processing

## Automatic Cropping Tissue Area

The first step is to find a tissue area automatically. Find improperly scanned images and ignore them.

Cropper output: an XML file.

<?xml version="1.0" encoding="ISO-8859-1"?>

<column id="0" width="2400">

<image name="20080423\_121806\_x148.0967y26.6286z10.6770\_t0.001000\_v20.9807">

<startx>937</startx>

<valid>true</valid>

</image>

<image name="20080415\_143127\_x148.0967y26.6286z10.6460\_t0.001000\_v17.5265">

<startx>-1</startx>

<valid>false</valid>

</image>

...

</column>

The second step is to actually crop image, remove noise, and normalize intensity level.

# Install tools

Prerequisites: Visual Studio 2008. Qt 4.7.4 seems not support Visual Studio 2010. Let us stick to this version.

## Git - Revision Control System

Revision control software. Git uses the distributed model. Each developer works with one's local repository. Changes are shared between repositories as a separate step.

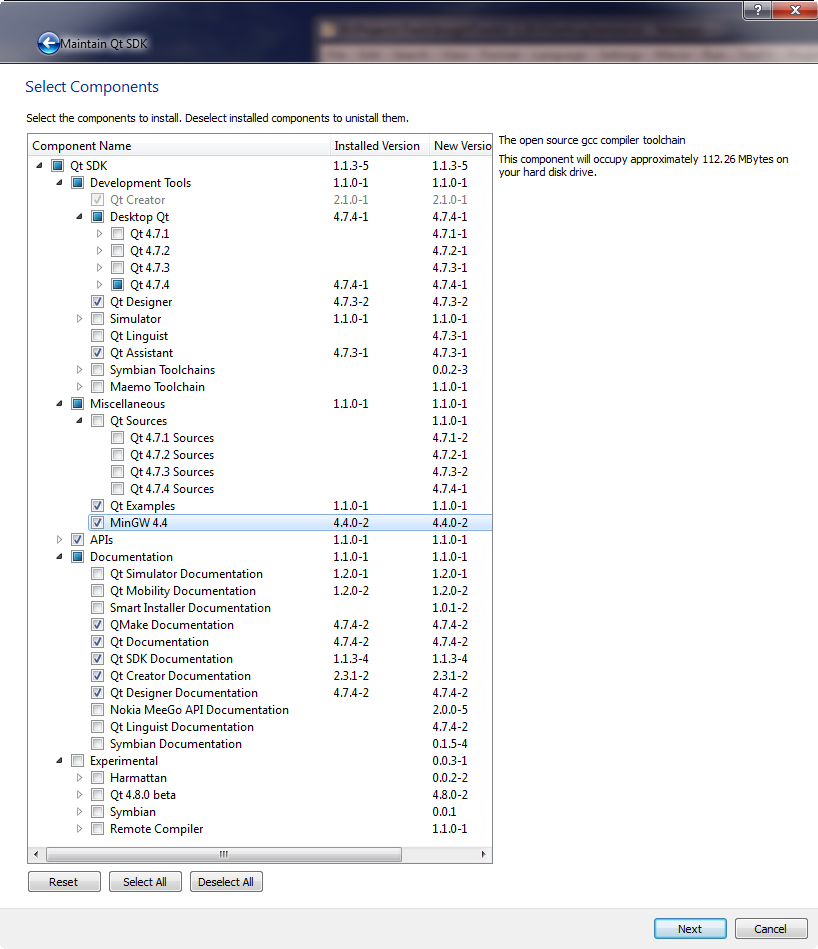
Download Full installer for official Git for Windows 1.7.7.

## Qt - UI Development tool

Download Qt, the cross-platform application framework. This will be a default IDE for KESM Suite.

QtSDK version 1.1.3 (Qt libraries version 4.7.4 and Qt Creator IDE version 2.3).

~~Make sure that you select MinGW that is an open source gcc compiler toolchain.~~



~~Add the path of MinGW into your system's PATH environment variable.~~

~~Find Advanced System Settings from your Control Panel (Windows 7). Click the Environment Variables button. Add PATH variable by clicking New.. button in 'User variables for XXX' section. Type '<Your Own Path of QtSDL>\mingw\bin' folder in the value.+~~

## wINDOWS sdk

Windows SDK installation. This is necessary to install Debugging tools for Windows. Also itk or vtk libraries need to link some of Windows libraries.

Install Windows-SDK Setup. Install to /Projects/Tools/**Microsoft-SDKs**/Windows/v7.1

Change the default path not to have a space.

**Note:** Visual Studio does not include the Debugging tools needed, and therefore, you must install them separately.

## CMake

CMake 2.8.

## ImageMagick

ImageMagick-6.7.3.0-Q8-windows-dll.exe

## InsightToolkit

### Building itk on Win32.

Download InsightToolkit-3.20.0.zip and unzip it.

Run CMake (cmake-gui). Select the folder as the source folder. Make a destination folder 'InsightToolkit-3.20.0**-build**.'

Click the Configure button. You will be asked 'Specify the generator for this project.' Let us choose Visual Studio 9 2008.

The destination folder for binary files can be set here. The default value is <Program Files Folder>\ITK. You may change this to your preferable location (/Projects/Tools/ITK). This location has all headers and lib files that you need when you build your own ITK applications.

Click the Generate button. It makes all necessary make files to create itk binary files.

Start Visual Studio 2008 and open ITK.sln. Build Solution by clicking F7.

### Install itk.

Find INSTALL target and select it. Right click.. Select Project Only - Build Only INSTALL. It will install all compiled binaries into the folder that you selected after configure.

## vtk

Download vtk-5.8.0.zip. Do not download vtk-5.8.0-win32-x86.exe (Windows Installer). This only has a binary executable file for Tcl/Tk. We need VTK libraries. We will use them to build VTK applications.

Unzip it.

Run CMake (cmake-gui). Select the unzipped folder as the source folder. Make a destination folder 'vtk-5.8.0-**build**.'

Click the Configure buttion and choose 'Visual Studio 9 2008.

Select VTK\_USE\_QT. This lead to set QT\_QMAKE\_EXECUTABLE. Set this with qmake.exe with full path name. (/Projects/Tools/QtSDK/Desktop/Qt/4.7.4/msvc2008/bin/qmake.exe.

Change the default folder of CMAKE\_INSTALL\_PREFIX to one that you want. (/Projects/Tools/VTK)

# Using itk with Qt

QtCreator is the main IDE for KESMSuite. This section explains how to use ITK libraries with Qt project files .

## Define Environment Variables

itk installation folder is too long. Let us define two environment variables to make the project file simpler.

A typical installation folder is K:\Projects\Tools\ITK. But include folder is in the deeper inside. K:\Projects\Tools\ITK\include\InsightToolkit. Libraries folder is also something like, K:\Projects\Tools\ITK\lib\InsightToolkit.

ITK\_INCLUDE\_DIR

ITK\_LIB\_DIR

In order to use ITK in your Qt Project file, you have to add INCLUDEPATH and LIBS.

INCLUDEPATH += $$(ITK\_INCLUDE\_DIR) \

$$(ITK\_INCLUDE\_DIR)/Common \

$$(ITK\_INCLUDE\_DIR)/Utilities \

$$(ITK\_INCLUDE\_DIR)/Utilities/vxl/vcl \

$$(ITK\_INCLUDE\_DIR)/Utilities/vxl/vcl/iso \

$$(ITK\_INCLUDE\_DIR)/Utilities/vxl/core \

$$(ITK\_INCLUDE\_DIR)/Utilities/vxl/core/vnl \

$$(ITK\_INCLUDE\_DIR)/Algorithms \

$$(ITK\_INCLUDE\_DIR)/BasicFilters \

$$(ITK\_INCLUDE\_DIR)/gdcm \

$$(ITK\_INCLUDE\_DIR)/IO \

$$(ITK\_INCLUDE\_DIR)/Numerics \

$$(ITK\_INCLUDE\_DIR)/SpatialObject \

$$(ITK\_INCLUDE\_DIR)

# itk libs

LIBS += $$(ITK\_LIB\_DIR)/ITKAlgorithms.lib \

$$(ITK\_LIB\_DIR)/ITKBasicFilters.lib \

$$(ITK\_LIB\_DIR)/ITKCommon.lib \

$$(ITK\_LIB\_DIR)/ITKDICOMParser.lib \

$$(ITK\_LIB\_DIR)/ITKEXPAT.lib \

$$(ITK\_LIB\_DIR)/ITKFEM.lib \

$$(ITK\_LIB\_DIR)/itkgdcm.lib \

$$(ITK\_LIB\_DIR)/ITKIO.lib \

$$(ITK\_LIB\_DIR)/itkjpeg12.lib \

$$(ITK\_LIB\_DIR)/itkjpeg16.lib \

$$(ITK\_LIB\_DIR)/itkjpeg8.lib \

$$(ITK\_LIB\_DIR)/ITKMetaIO.lib \

$$(ITK\_LIB\_DIR)/itkNetlibSlatec.lib \

$$(ITK\_LIB\_DIR)/ITKniftiio.lib \

$$(ITK\_LIB\_DIR)/ITKNrrdIO.lib \

$$(ITK\_LIB\_DIR)/ITKNumerics.lib \

$$(ITK\_LIB\_DIR)/itkopenjpeg.lib \

$$(ITK\_LIB\_DIR)/itkpng.lib \

$$(ITK\_LIB\_DIR)/ITKSpatialObject.lib \

$$(ITK\_LIB\_DIR)/ITKStatistics.lib \

$$(ITK\_LIB\_DIR)/itksys.lib \

$$(ITK\_LIB\_DIR)/itktiff.lib \

$$(ITK\_LIB\_DIR)/itkv3p\_lsqr.lib \

$$(ITK\_LIB\_DIR)/itkv3p\_netlib.lib \

$$(ITK\_LIB\_DIR)/itkvcl.lib \

$$(ITK\_LIB\_DIR)/itkvnl.lib \

$$(ITK\_LIB\_DIR)/itkvnl\_algo.lib \

$$(ITK\_LIB\_DIR)/itkvnl\_inst.lib \

$$(ITK\_LIB\_DIR)/itkzlib.lib \

$$(ITK\_LIB\_DIR)/ITKznz.lib

# WIN32 libs

LIBS += "K:\Projects\Tools\Microsoft-SDKs\Windows\v7.1\Lib\AdvAPI32.Lib"

Do not forget to add WIN32 libs. You would see complains link errors from your itksys.lib.

Issue 1: No clear right edge.

The arrow indicates the true right edge. But it is hard to identify it from the automatic tissue area detector.

Destination folder structure

\KESMData\<RawDataFolderName>\

\KESMData\<RawDataFolderName>\RawDataInfo.xml

\KESMData\<RawDataFolderName>\Template.jpg

\KESMData\<RawDataFolderName>\Cropped\<OrgColNo>.xml

\KESMData\<RawDataFolderName>\Cropped\< OrgColNo>\org\_name.jpg

\KESMData\<RawDataFolderName>\Relighted\<OrgColNo>\org\_name.jpg

\KESMData\<RawDataFolderName>\Merged\<NewlyIndexed>.jpg

\KESMData\<RawDataFolderName>\Tiled\<folder structure for Google Maps API>

\KESMData\<RawDataFolderName>\Scaled\<Size>\<NewColNo>\<NewlyIndexed>.jpg

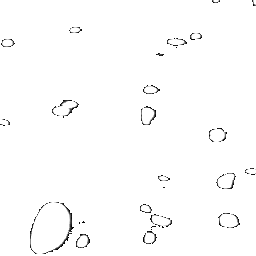
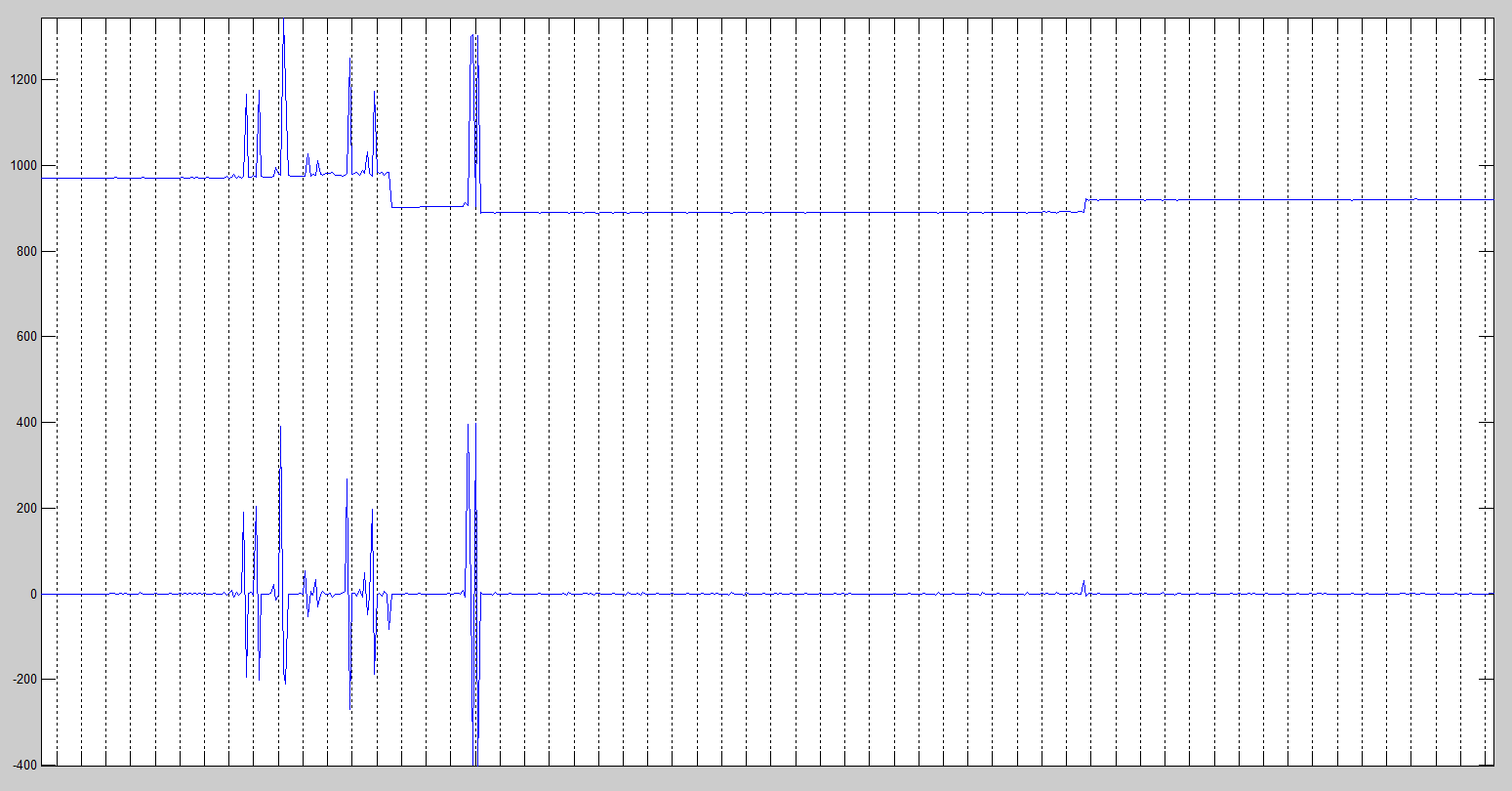
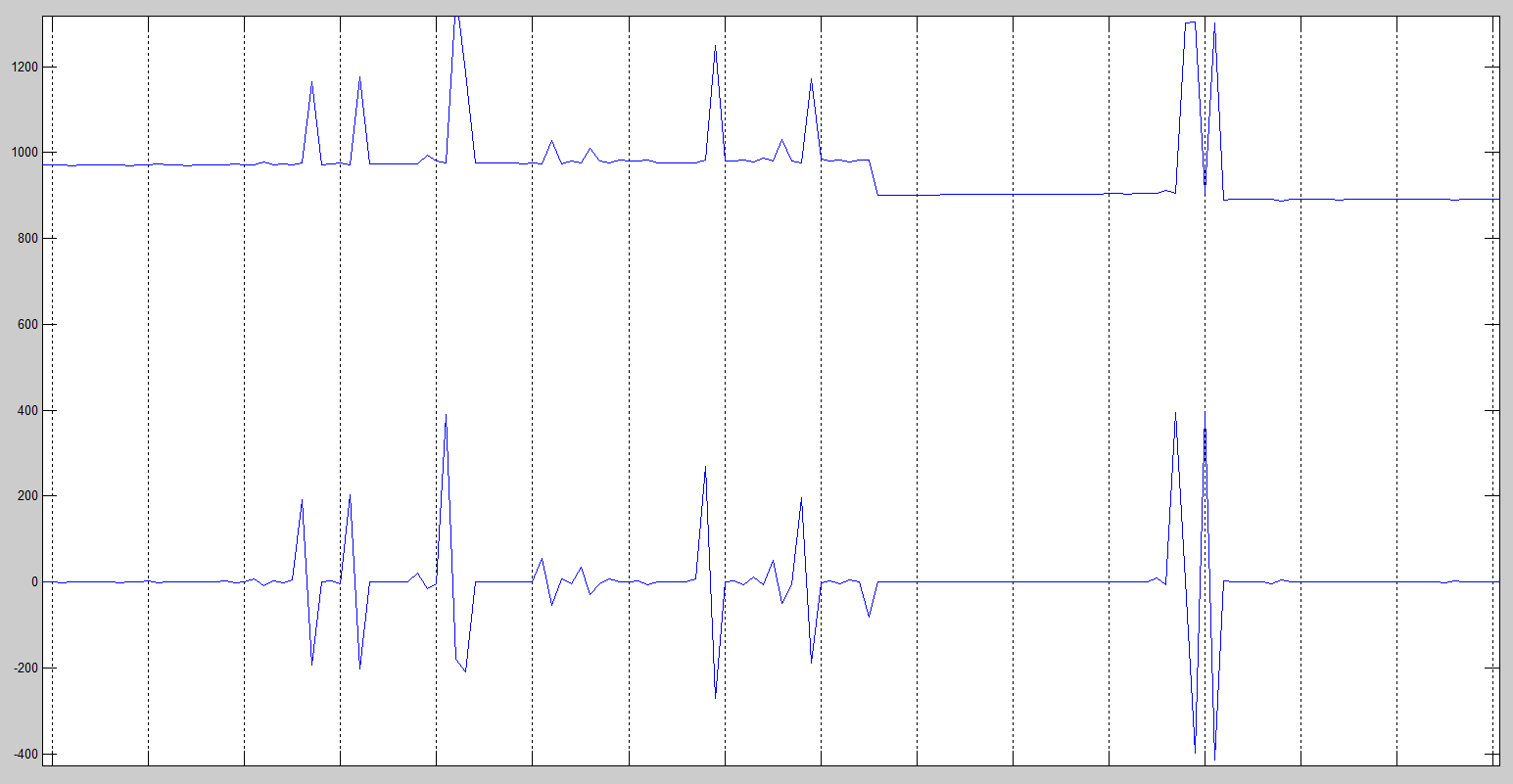
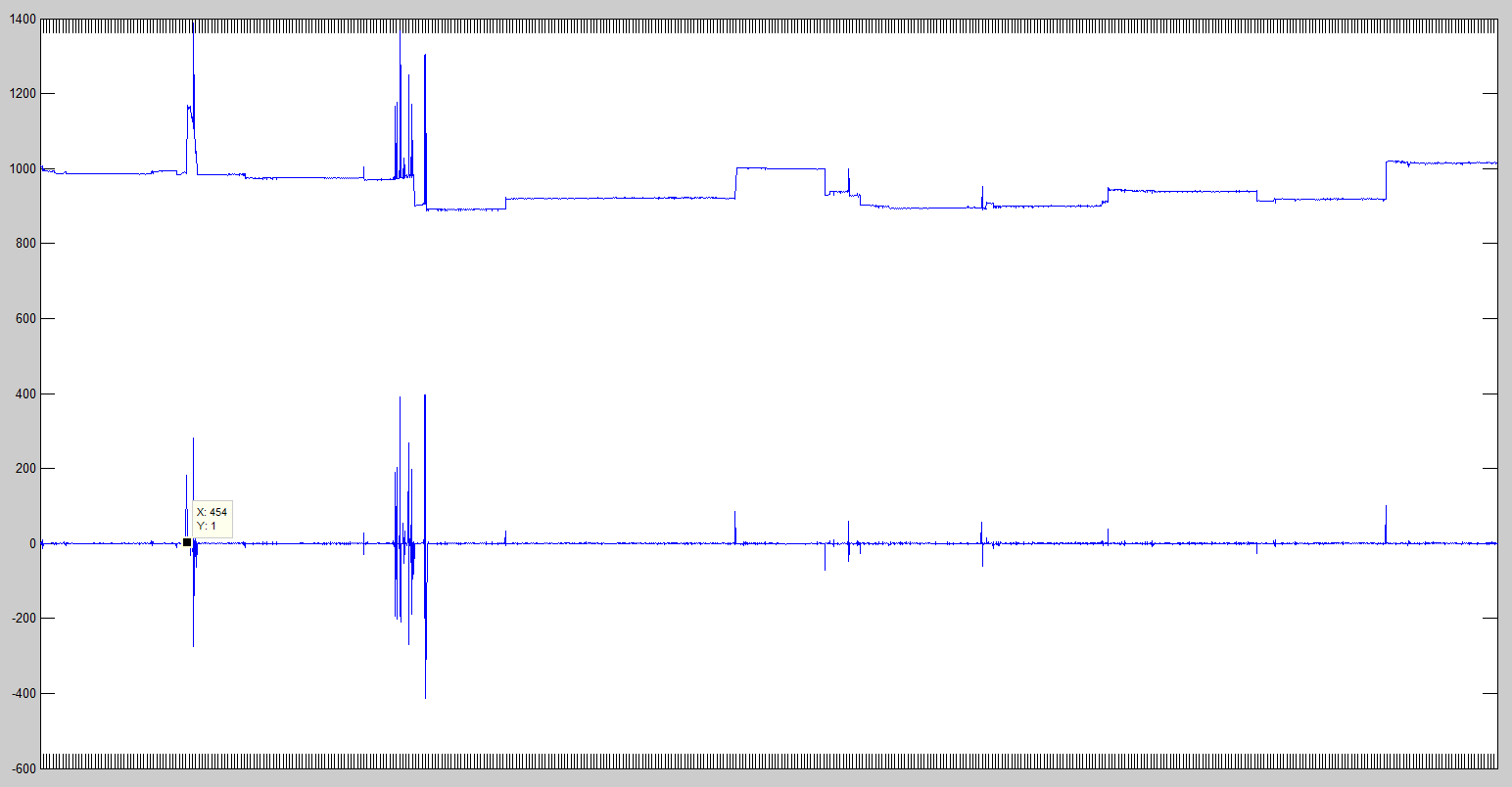
 

Figure 1. The current image in the KESM brain atlas Figure 2. The new image

In the KESM brain atlas, the tissue background of images is not completely transparent due to uneven normalization (see Figure 1). Before making tissue background transparent, we should normalize the intensity level of an image first. Figure 2 shows that the tissue background is completely transparent from an normalized image.







x-axis: Raw KESM images in a column.

Upper graph: Start X positions of the tissue area.

Lower graph: Difference of two neighboring images in Start X positions.

In the lower graph, sharp and one way (either upward or downward) spikes mean the start of a new chunk. Multiple oscillating spikes (up and down frequently) mean outliers that do not have a clear right edge that can be detected by the automatic algorithm but most likely majority of the images have meaningful tissue information.

