

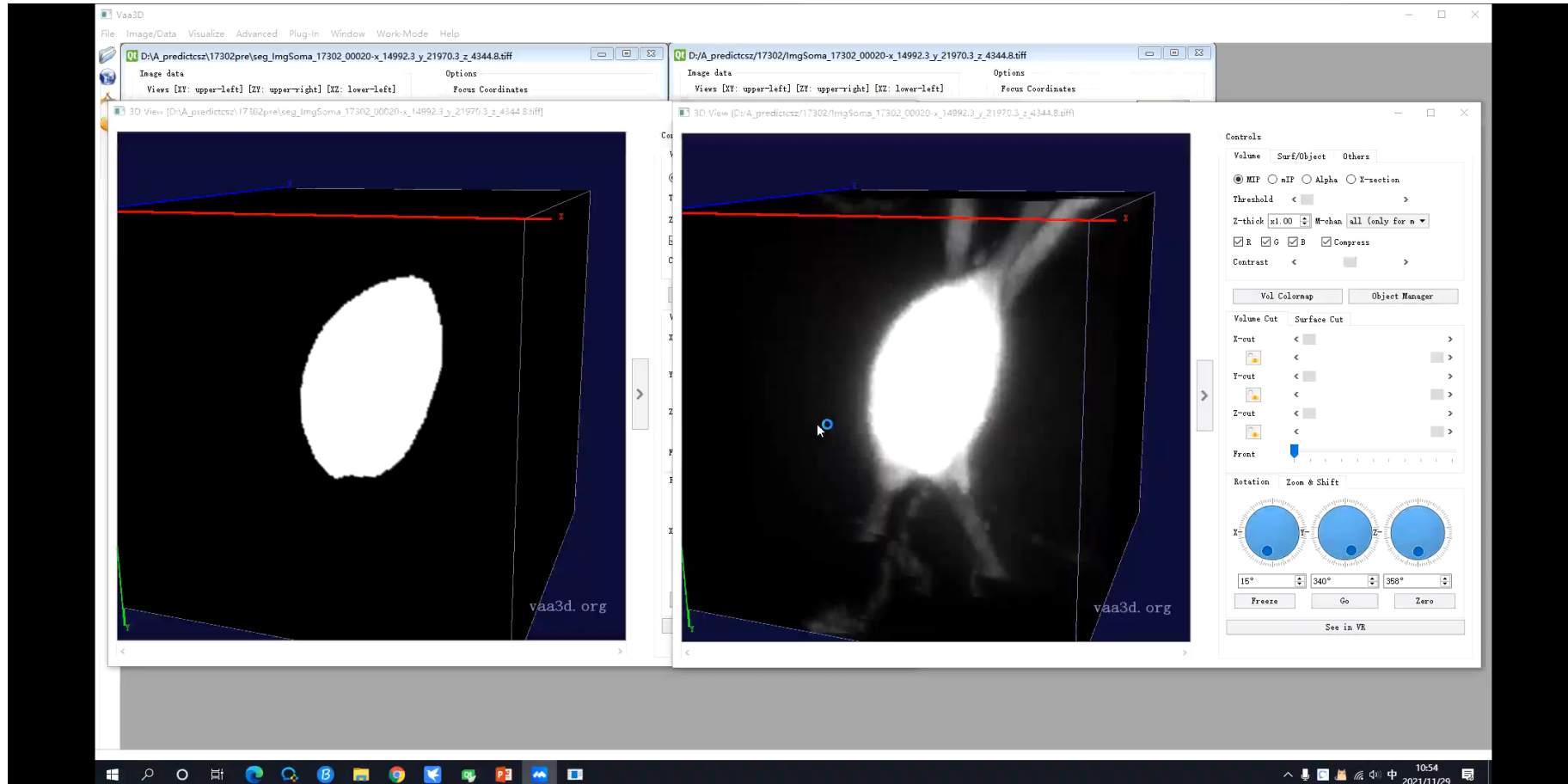
Create a plugin to mix images

Chen Shize

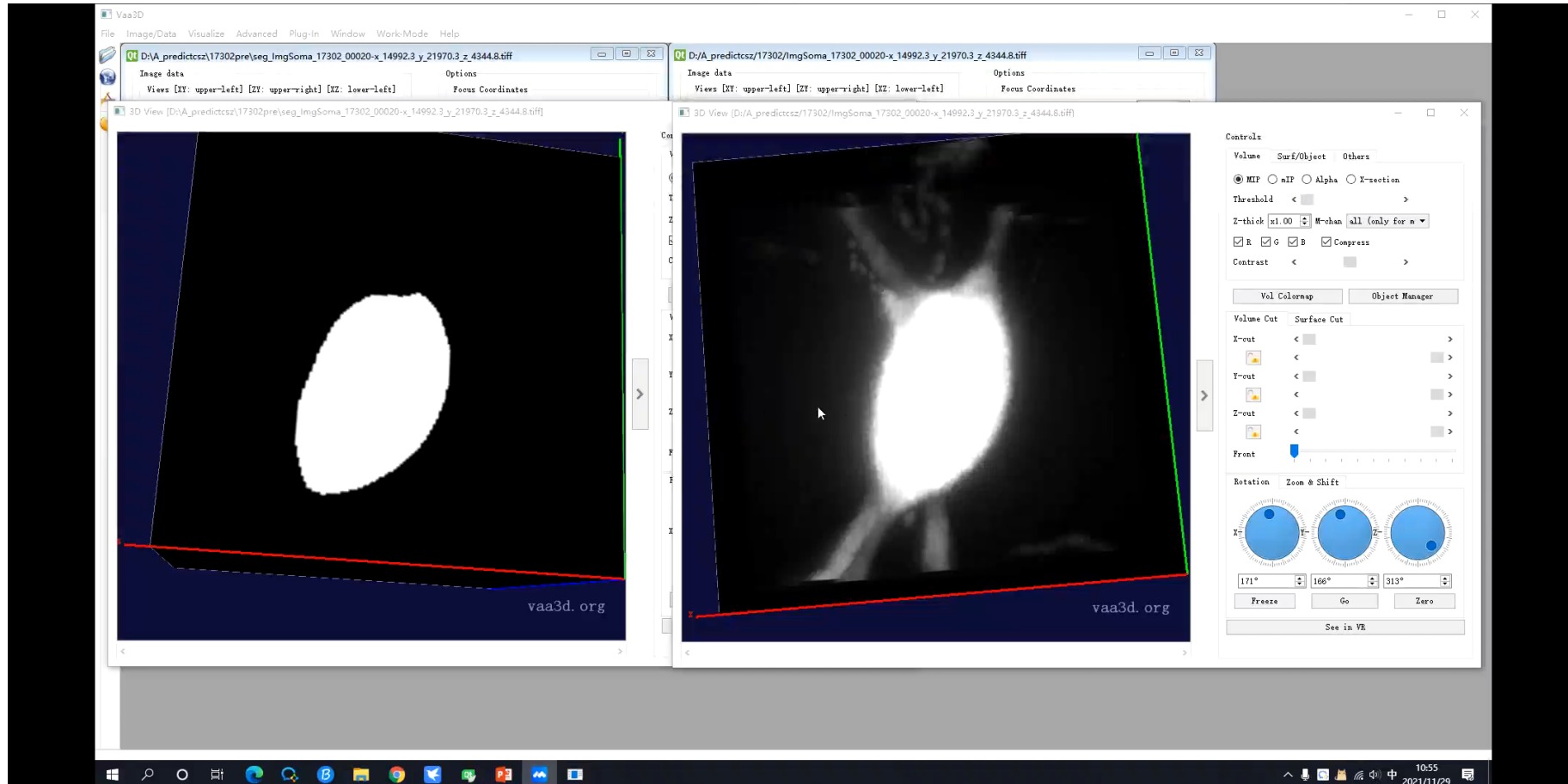
Reference

- One example of creating a plugin(image threshold):
- https://github.com/Vaa3D/Vaa3D_Wiki/wiki/PluginDesignGuide.wiki
- PluginAPI:
- https://github.com/Vaa3D/Vaa3D_Wiki/wiki/PluginAPI.wiki

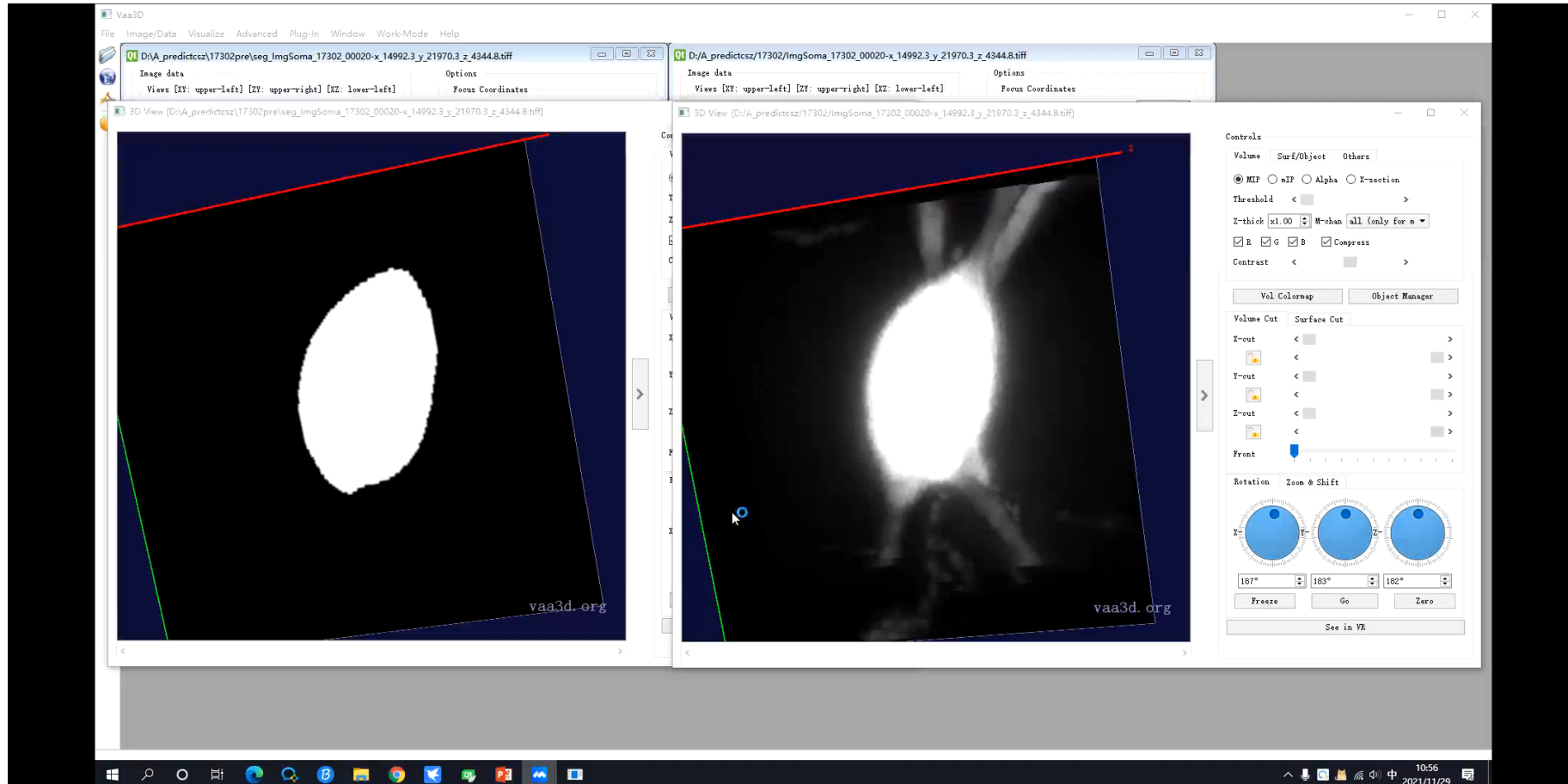
Why do I need to mix images



Why do I need to mix images



Why do I need to mix images



Reason

- 1. Dragging the images randomly will lead to the different observation angle of view.
- 2. When using the Rotation UI in V3D's 3Dviewer, we may not be able to observe the images from the viewing angle we want.
- 3. If we want to implement the same operation in the two images, we need to take full control of 3DViewer, which is not safe.

The preparatory work

- V3d source code
- Qt 4.8.6
- Qt creator(or other IDE)
- [PluginAPI.wiki](#)
- [PluginDesignGuide.wiki](#)

The algorithm

- Read and load image1,image2
- $\text{Image3} = \text{image1} * \alpha + \text{image2} * (1 - \alpha)$
- Show image3

We need two images to mix and one to save, which means we need to open three images in v3d_main window(tri-view windows). The first two images are the images we want to mix and the third image is load as a window to save the mixed result.

Load the images

```
const QString title="Image Thresholding";
int image_threshold(V3DPluginCallback2 &callback, QWidget *parent)
{
    // 1 - Obtain the current 4D image pointer
    v3dhandle curwin = callback.currentImageWindow();
    if(!curwin)
    {
        QMessageBox::information(0, title, QObject::tr("No image is open.));
        return -1;
    }
    Image4DSimple *p4DImage = callback.getImage(curwin);
}
```

- virtual v3dhandle currentImageWindow()=0;
- virtual Image4DSimple* getImage(v3dhandle image_window)=0;

PluginAPI(get opened images)

- v3dhandleList *getImageWindowList()*
- v3dhandle *currentImageWindow()*
- v3dhandle *curHiddenSelectedWindow()*

- v3dhandleList is a list of the v3dhandle pointers, which points to the tri-view windows.
- If only one image opened:
 - `callback.getImageWindowList()[0]=callback.currentImageWindow()`

First step: obtain the data structure

```
v3dhandleList curlist=callback.getImageWindowList();

if(curlist.size()<3){
    QMessageBox::information(0, title, QObject::tr("You need at least three images, the last image is needed as the result stack."));
    return -1;
}

v3dhandle win1=curlist[0];
v3dhandle win2=curlist[1];
v3dhandle win3=curlist[2];
Image4DSimple *p4DImage1 = callback.getImage(win1);
Image4DSimple *p4DImage2 = callback.getImage(win2);
Image4DSimple *p4DImage3 = callback.getImage(win3);
```

- *p4DImage is a pointer which points to the data structure of the image.

Second step: traverse the matrix

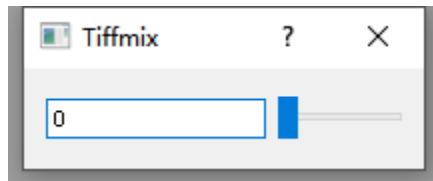
- The data matrix is one-dimensional

```
V3DLONG sz[3];
    sz[0] = p4DImage->getXDim();
    sz[1] = p4DImage->getYDim();
    sz[2] = p4DImage->getZDim();
V3DLONG tb = sz[0]*sz[1]*sz[2]*p4DImage->getUnitBytes();
for (V3DLONG i=0;i<tb;i++)
```

Third step: mix the images

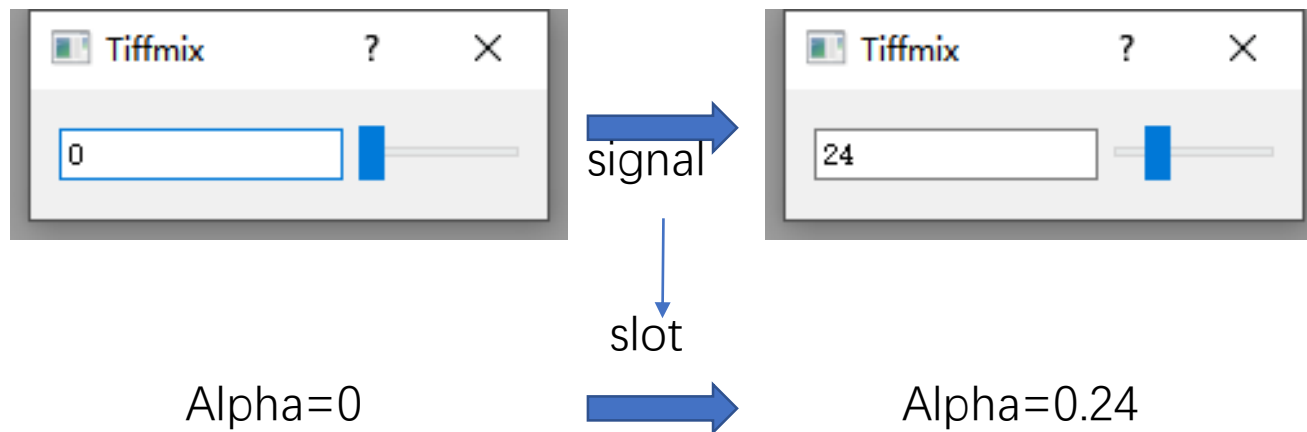
- Problems:
- 1. We cannot change the UI of 3DViewer.
- 2. We hope the alpha value can be changed as we want.

We need the UI to change the alpha value.



Signal transmission

- `connect(sender,SIGNAL(signal()),receiver,SLOT(slot()));`
- `Signal()` is defined to send the signal.
- `Slot()` is defined to get the signal.



Recompute when alpha changed

- Problems:
- We want to display our UI through the plugin.
- We need the class `V3DPluginCallback2` to obtain and show the image, but the slot function can only get the signal, this means we need rerun the mix-function in the slot function with only one parameter(the signal).

Announce a UI class in the plugin class, and define a pointer to obtain the `V3DPluginCallback2` Object.

Fourth step: show the mixed image

```
v3dhandle newwin = callback.newImageWindow();  
p4DImage->setData(nm, sz[0], sz[1], sz[2], 1, p4DImage->getDatatype());//setData() will free the original memc  
callback.setImage(newwin, p4DImage);  
callback.setImageName(newwin, QObject::tr("Image Thresholding"));  
callback.updateImageWindow(newwin);
```

← A new tri-view window will be generated

WARNING: More and more images will be generated!

Fourth step: show the mixed image

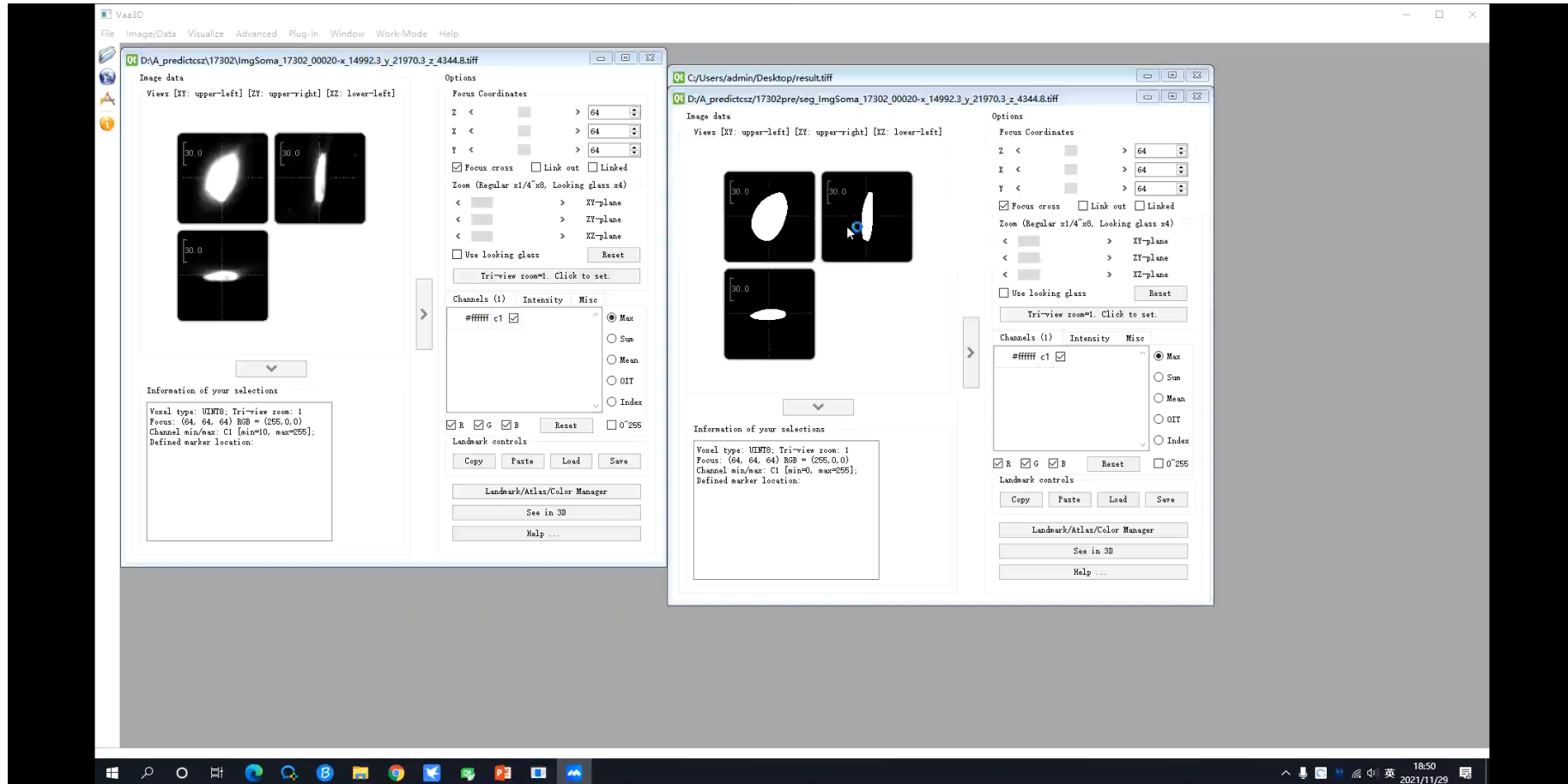
- `//update the content in a 3D viewer directly`
- `virtual void pushImageIn3DWindow(v3dhandle image_window) = 0;`
- **The program will crash if the image is pushed too many times.**

`//open and close a global 3D viewer`

`virtual void open3DWindow(v3dhandle image_window) = 0;`

`virtual void close3DWindow(v3dhandle image_window) = 0;`

Fourth step: show the mixed image



Thank you!