VTK学习（十二）图像切面提取

原创 yuyanjia 2017-11-13 21:46:54 4180 收藏 4
分类专栏： VTK

```
/*基于VTK9.0和VS2019/VS7.1. By yyy 2017.11.13*/


#include "vtkAutoInit.h"
VTK_MODULE_INIT(vtkRenderingOpenGL2); // VTK was built with vtkRenderingOpenGL2
VTK_MODULE_INIT(vtkInteractionStyle);
#include vtkImagePlaneWidget.h
#include vtkImageReader2.h
#include vtkMetaImageReader.h
#include vtkImageResliceCast.h
#include vtkImageMapToColors.h
#include vtkImageMapTo1Colors.h
#include vtkImageActor.h
#include vtkRenderer.h
#include vtkRenderWindow.h
#include vtkRenderWindowInteractor.h
#include vtkInteractorStyleImage.h
#include vtkCommand.h
#include vtkImageData.h
#include vtkMetaImageReader.h
#include vtkImageCast.h


class vtkImageInteractionCallback : public vtkCommand
{
public:
    static vtkImageInteractionCallback *New()
    {
        return new vtkImageInteractionCallback;
    }

    vtkImageInteractionCallback()
    {
        this->Slicing = 0;
        this->ImageReslice = 0;
        this->Interactor = 0;
    }

    void SetImageReslice(vtkImageReslice *reslice)
    {
        this->ImageReslice = reslice;
    }


    void SetImageMapToColors(vtkImageMapToColors *mapToColors)
    {
        this->mapToColors = mapToColors;
    }

    vtkImageReslice *GetImageReslice()
    {
        return this->ImageReslice;
    }

    void SetInteractor(vtkRenderWindowInteractor *Interactor)
    {
        this->Interactor = interactor;
    }

    vtkRenderWindowInteractor *GetInteractor()
    {
        return this->Interactor;
    }

    virtual void Execute(vtkObject *, unsigned long event, void *)
    {
        vtkRenderWindowInteractor *Interactor = this->GetInteractor();

        int lastPos[2];
        int currPos[2];
        interactor->GetLastEventPosition(lastPos);
        int currPos[2];
        interactor->GetEventPosition(currPos);

        if (event == vtkCommand::LeftButtonPressEvent)
        {
            this->Slicing = 1;
        }
        else if (event == vtkCommand::LeftButtonReleaseEvent)
        {
            this->Slicing = 0;
        }
        else if (event == vtkCommand::MouseMoveEvent)
        {
            if (this->Slicing)
            {
                vtkImageReslice *reslice = this->ImageReslice;

                // Increment slice position by deltaY of mouse
                int deltaY = lastPos[1] - currPos[1];

                reslice->Update();
                double sliceSpacing = reslice->GetOutput()->GetSpacing()[2];
                vtkMatrix4x4 *matrix = reslice->GetResliceAxes();
                // move the center point that we are slicing through
                double point[4];
                double center[4];
                point[0] = 0.0;
                point[1] = 0.0;
                point[2] = sliceSpacing * deltaY;
                point[3] = 1.0;
                matrix->MultiplyPoint(point, center);
                matrix->SetElement(0, 3, center[0]);
                matrix->SetElement(1, 3, center[1]);
                matrix->SetElement(2, 3, center[2]);
                mapToColors->Update();
                interactor->Render();
            }
            else
            {
                vtkInteractorStyle *style = vtkInteractorStyle::SafeDownCast(
                    interactor->GetInteractorStyle());
                if (style)
                {
                    style->OnMouseMove();
                }
            }
        }
    }

private:
    int Slicing;
    vtkImageReslice *ImageReslice;
    vtkRenderWindowInteractor *Interactor;
    vtkImageMapToColors *mapToColors;
};

int main()
{
    vtkMetaImageReader* reader = NULL;
    reader = vtkSmartPointer<vtkMetaImageReader>::New();
    reader->SetFileName("E:/first_year_graduate/VTK/samples/Examples/Examples/Cha
    reader->Update();

    int extent[6];
    double spacing[3];
    double origin[3];

    reader->GetOutput()->GetExtent(extent);
    reader->GetOutput()->GetSpacing(spacing);
    reader->GetOutput()->GetOrigin(origin);

    double center[3];
    center[0] = origin[0] + spacing[0] * 0.5 * (extent[0] + extent[1]);
    center[1] = origin[1] + spacing[1] * 0.5 * (extent[2] + extent[3]);
    center[2] = origin[2] + spacing[2] * 0.5 * (extent[4] + extent[5]);

    static double axialElements[16] = {
        1, 0, 0, 0,
        0, 1, 0, 0,
        0, 0, 1, 0,
        0, 0, 0, 1
    };

    vtkSmartPointer<vtkMatrix4x4> resliceAxes =
        vtkSmartPointer<vtkMatrix4x4>::New();
    resliceAxes->DeepCopy(axialElements);

    resliceAxes->SetElement(0, 3, center[0]);
    resliceAxes->SetElement(1, 3, center[1]);
    resliceAxes->SetElement(2, 3, center[2]);

    vtkSmartPointer<vtkImageReslice> reslice =
        vtkSmartPointer<vtkImageReslice>::New();
    reslice->SetInputConnection(reader->GetOutputPort());
    reslice->SetOutputDimensionality(2);
    reslice->SetResliceAxes(resliceAxes);
    reslice->SetInterpolationModeToLinear();

    vtkSmartPointer<vtkLookupTable> colorTable =
        vtkSmartPointer<vtkLookupTable>::New();
    colorTable->SetRange(0, 1000);
    colorTable->SetValueRange(0.0, 1.0);
    colorTable->SetHueRange(0.0, 0.0);
    colorTable->SetSaturationRange(0.0, 0.0);
    colorTable->Build();

    vtkSmartPointer<vtkImageMapToColors> colorMap =
        vtkSmartPointer<vtkImageMapToColors>::New();
    colorMap->SetLookupTable(colorTable);
    colorMap->SetInputConnection(reslice->GetOutputPort());
    colorMap->Update();

    vtkSmartPointer<vtkImageActor> imgActor =
        vtkSmartPointer<vtkImageActor>::New();
    imgActor->SetInputData(colorMap->GetOutput());

    vtkSmartPointer<vtkRenderer> renderer =
        vtkSmartPointer<vtkRenderer>::New();
    renderer->AddActor(imgActor);
    renderer->SetBackground(1, 1, 1);

    vtkSmartPointer<vtkRenderWindow> renderWindow =
        vtkSmartPointer<vtkRenderWindow>::New();
    renderWindow->SetSize(300, 300);
    renderWindow->AddRenderer(renderer);

    vtkSmartPointer<vtkRenderWindowInteractor> renderWindowInteractor =
        vtkSmartPointer<vtkRenderWindowInteractor>::New();
    vtkSmartPointer<vtkInteractorStyleImage> imageStyle =
        vtkSmartPointer<vtkInteractorStyleImage>::New();

    renderWindowInteractor->SetInteractorStyle(imageStyle);
    renderWindowInteractor->SetRenderWindow(renderWindow);
    renderWindowInteractor->Initialize();

    vtkSmartPointer<vtkImageInteractionCallback> callback =
        vtkSmartPointer<vtkImageInteractionCallback>::New();
    callback->SetImageReslice(reslice);
    callback->SetInteractor(renderWindowInteractor);
    callback->SetImageMapToColors(colorMap);

    imageStyle->AddObserver(vtkCommand::MouseMoveEvent, callback);
    imageStyle->AddObserver(vtkCommand::LeftButtonPressEvent, callback);
    imageStyle->AddObserver(vtkCommand::LeftButtonReleaseEvent, callback);

    renderWindowInteractor->Start();
    return EXIT_SUCCESS;
}
```

运行结果：

按下鼠标左键，移动鼠标时的情形：

运行结果：

按下鼠标左键，移动鼠标时的情形：

代码解释：

1. 先通过vtkMetaImageReader读取一基三维图像，获取图像范围、原点和像素间隔，由这三个参数可以计算图像的中心位置。
2. 接下来定义了切面的变换矩阵axialElements，该矩阵的前三列分别表示X、Y和Z方向矢量，第四列为切面坐标系原点，通过修改切面坐标系原点，可以得到不同位置的切面图像。
3. 然后将该矩阵设置给了vtkImageReslice的输入，通过函数SetResliceAxes()设置变换矩阵resliceAxes。

转载自http://blog.csdn.net/webzhuce/article/details/69178566

基于VTK的医学三维图像切割提取与可视 07-12
一款完整的基于开源软件的医学三维图像切割工具。基于面向对象的设计思想，以及模块化的设计思想，使用...

基于VTK的任意平面切割 481
从切割方向不对于一般情况的切割图像来说必须，相等的切片实现切割，但是需要有一个模型，然后再一个切割平...

暂无评论内容，发表第一条评论吧！ 评论

相关小程序：想好，游戏!现在已知四面体的四个顶点坐标分别为P0(x,0,0)P1(x,y,0)P2(x,y,z)P3(x,0,z)求四个 8 月前 回复 ···

哈哈哈哈哈哈小程序：我想回复下知道怎么回复了？在PROR: in d:\programewar\vtk-8.1.0\vtk-8.1.0\vendor\vtkmetaimagereader\vtkmetaimagereader.cxx, line 237 vtkMetaImageReader (0000015D02C28790): Metaimage cannot parse file. ERROR: in d:\programewar\vtk-8.1.0\vtk-8.1.0\vendor\vtkmetaimagereader.cxx, line 237 vtkMetaImageReader (0000015D02C28790): Metaimage cannot read data from file. 可以给出解决问题的吗？ 3 年前 回复 ···

diaoshang001 回复 : [replyjbuanbanyq]哈哈哈哈，请问你回复的不了吗？ 3 年前 回复 ···

哈哈哈哈哈小程序哈哈 回复 : 当然我知道你是想问题是"D:\programe\TKindofbrain.4brain.nhd" 3 年前 回复 ···

相关推荐

VTK虚拟切片提取 OneSeal的专栏 © 3791
参考资料 (1)VTK学习笔记之TIFF图像数据的读写(1) ...

VTK_Learning_三维图像切片（一） ...wxm_36334343的博客 © 758
1、三维图像切片提取 切片实质：把图像分成一个个切面的切面图像，切面可以是任意图像内部一点任平行于X Y、YZ、...

VTK中切片提取方法 qq_42779015的博客 © 1254
目前遇到一个要求，要提取到VTK的二维切面数据并对其进行切片提取，利用到的数据是vtkImageReslice,下面是该类的一个...

从三维数据中提取一定结果二维数据，并保存在新的二维数据中 09-07
我的问题是：三维数据中提取的二维数据，并保存在新的二维数据中

VTK 多平面重建（MPR）及三维切片显示 ith_7_1024的博客 © 15k+
在医学图像数据分析时，两个最常用的工具是多平面切片（MPR）和三维表面重建（3D）。MPR是指沿着三个正交平面（轴状、冠状和矢状）切割...

VTK 三维 重建 体绘制 面绘制 截面 平面切割 东北日报社 © 1785
VTK 三维 重建 体绘制 面绘制 截面 平面切割 最初使用MFC的，VTK的VRC++，平台无关，一个人只有研究...

VTK序列图像的读取 东溟工作室 © 9846
我可图像处理的应用程序中，经常会遇到如下一个序列图像的操作，比如DICT，MRI等等的图像数据，一个切面一个...

VTK：对输入的三维模型在某个方向等间距切割模型的切面数据集 h5011470的博客 © 3276
测试环境：操作系统Ubuntu，版本号18.04 ，编译器版本 7 ，编译工具 cmake ，vtk版本vtk9.0.0 #include <vtkSmartPointer.h> #include <vtkSmartPointer.h> ...

OpenCV：图像切片 sinoir © 2425
#include <opencv2/opencv.hpp> #include <opencv2/imgproc/imgproc.hpp> #include <opencv2/highgui/highgui.hp...

VTK移除之虚拟？图像基本操作，三维图像切片交互显示（回调函数）。 交易猫的博客 © 3302
1 创建自定义事件，三维图像基本操作，三维图像切片交互显示（回调函数）。 2 在交互类一个初始化函数中，一般在显示函数...

Study-VTK：图形图像域 小叔的博客 © 649
无基本书 vtk教程文档 官网源码 搜索模板vtk模型的解决方案 通域之声 自定义图形 自定义图形 ...

matlab实现对图像的切割 zhouzhihua © 273+
在四维图像中取一个切面的方法，使常常需要将图像转化成一个切面的二维图像，这个切面的方法是可以任意的，imgs=reslice...

VTK笔记——空间几何变换(Transform)，平移、旋转和缩放 Andy的笔记 © 1.1k+
先做一组图，下图是一个三维的Cube，试图把旋转中，放大，放小，缩放的操作做了一遍。1.RlCube=新建对象类型 2.Cube...

VTK图像处理之vtkImageReslice 青岛小鸟的博客笔记 © 8163
三维图像切面提取的实现，vtkImageReslice 三维图像切面提取的实现，vtkImageReslice 三维图像切面提取的实现，vtkImageReslice