

基于 Matlab 的脑部 MRI 图像三维重建与动画显示

戴虹

(上海第二工业大学电子电气工程学院, 上海 201209)

摘要: 基于 Matlab 软件编程实现 MC 算法对人体脑部 MRI 图像进行面绘制三维重建; 利用三维旋转变换和动画函数对脑部图形进行动画显示。实验结果表明, 该算法能准确地对脑部进行全方位显示, 运行速度快且占的内存空间较少。

关键词: Matlab; 三维重建; 三维旋转变换; 动画

3D Reconstruction and Animation Display of Brain MRI Images Based on Matlab

DAI Hong

(School of Electronic and Electrical Engineering, Shanghai Second Polytechnic University, Shanghai 201209)

Abstract: The following algorithms are proposed and realized by Matlab programming based on the brain MRI images. The 3D surfaces of the brain is reconstructed using MC algorithm; A rotate animation of the brain is created and displayed by 3D rotate transformation and animation functions of Matlab. Result shows that the algorithm can show the brain accurately and quickly, takes up less space in memory.

Key words: Matlab; 3D Reconstruction; 3D Rotation Transformation; Animation

1 引言

计算机断层扫描 (CT) 及磁共振 (CT) 成像技术已广泛应用于疾病诊断中, 但这些成像手段只能提供人体内部的二维图像。近年来, 三维可视化技术已成为医学图像处理的研究热点。该技术能从断层图像序列中提取人体组织的三维结构信息, 在辅助诊断、手术仿真和引导治疗方面发挥重要作用。Matlab 是美国 MathWorks 公司出品的数学软件, 其功能十分强大, 用于几乎所有工程领域的算法开发和数据分析, 且编程简单、易学易用。

基于 Matlab 软件编程实现脑部 MRI 图像的三维重建并产生旋转动画, 可应用于脑部疾病诊断与医学领域的计算机辅助教学系统中。

2 三维重建

采用移动立方体法 (Marching Cubes, MC) 算法, 对脑部 MRI 图像进行面绘制三维重建。其基本思想是^[1]: 从三维数据场中根据物体表面特征的信息, 抽取每个体素的“等值面”, 由这些等值面组成物体的三维表面。步骤如下:

(1) 等值面抽取

将脑部图像序列按顺序叠加成一个三维体数据集 $f(x, y, z)$, “等值面”实际上是指空间中的一张曲面, 在该曲面上 $f(x, y, z)$ 等于值 k 。即

$$f(x, y, z) = k \quad (1)$$

其中 k 为域值。 k 取分割脑部图像时的阈值, $k=20$ 。为产生光滑表面, 在抽取之前用高斯低通滤波器对数据集进行平滑处理。

(2) 灯光效果设置

采用 phong 模型进行灯光效果设置。

3 脑部旋转动画显示

为了从各个方向观察脑部结构, 可对重建结果图形进行三维旋转变换, 并生成动画进行显示。基本思想是: 图形经过三维旋转, 由于其等值面连接方式不变, 因而结构不会发生变形。脑部图形绕其质心旋转及动画显示的步骤如下:

(1) 将图形的质心移至原点。齐次变换矩阵^[2]为:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_g & -y_g & -z_g & 1 \end{bmatrix} \quad (2)$$

$$\text{其中 } x_g = \frac{\sum_{i=1}^n x_i}{N}; y_g = \frac{\sum_{i=1}^n y_i}{N}; z_g = \frac{\sum_{i=1}^n z_i}{N}; N \text{---数据集像素}$$

总数; (x_i, y_i, z_i) ---数据集中像素 i 的坐标。

(2) 相对于原点 (即质心) 作旋转变换, 旋转变换矩阵为:

$$T_2 = R_x R_y R_z \quad (3)$$

$$R_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中

基金项目: 上海市教育委员会重点学科项目: 测控自动化 (J51801); 上海第二工业大学校重点课程建设: 《数字信号处理》课程建设 (KC0901)。

作者简介: 戴虹 (1977-), 女, 讲师, 在职博士生, 研究方向: 医学图像与信号处理。

收稿日期: 2010-07-06

$$R_Y = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_Z = \begin{bmatrix} \cos r & \sin r & 0 & 0 \\ -\sin r & \cos r & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

α, β, γ 是图形分别绕 X, Y, Z 轴旋转的角度。

(3) 进行平移变换, 将质心移至原来的位置, 变换矩阵为:

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_g & y_g & z_g & 1 \end{bmatrix} \quad (4)$$

综合以上 3 步, 脑部图形绕自身质心旋转的变换矩阵为:

$$T = T_1 \cdot T_2 \cdot T_3 \quad (5)$$

设图形上任意一点 (x, y, z) 绕自身的质心旋转后新坐标为 (x_1, y_1, z_1) , 则

$$[x_1, y_1, z_1, 1] = [x, y, z, 1] \cdot T \quad (6)$$

(4) 动画显示及存储

设脑部图形从原始位置开始, 每次以其质心为原点, 分别绕 X, Y, Z 轴旋转 $\Delta\alpha, \Delta\beta$ 和 $\Delta\gamma$ 角度, 显示并存储该幅画面, 作为动画文件的一帧, 在显示当前图形的同时擦除前一幅图形, 共旋转一周以产生动画效果^[3], 最终将所有画面存入 avi 格式的动画文件中, 从而能脱离 Matlab 环境播放。

4 Matlab 程序

编程环境: Matlab2007。实验数据来源: Matlab 自带的脑部 MRI 图像序列: "MRI"。

脑部的三维重建及旋转动画显示程序 (brainavi.m) 如下:

```
% brainavi.m
clear %清除内存
clc %清除屏幕
Figwin=figure('position',[50 50 450 450], 'Name',
'脑部三维重建及旋转动画演示', ...
'NumberTitle','off','MenuBar','none');
%产生标题为" 脑部三维重建及旋转动画演示" 的图形窗
%口
%%%1.读入脑部 MRI 图像%%%
load mri %调入脑部 MRI 图像数据
D=squeeze(D); %将 D 从 4 维转换为 3 维
Ds=smooth3(D); %采用高斯低通滤波器对 D 进行平
%滑, 得 Ds
%%%2.脑部三维重建与旋转动画显示%%%
fv=isosurface(Ds,20);
%脑部等值面抽取, 阈值 k=20, 见公式 (1)。fv 是一个结构
%数组, 其中
%fv.vertices 为图形的顶点信息; fv.faces 为图形的表面信
```

%息

```
fv2=isoscaps(D,5); %脑盖上盖的等值面抽取, 阈值 k=5
yuan=fv.vertices; %令 yuan 为原脑部图形的顶点信息
yuan2=fv2.vertices; %令 yuan2 为原脑盖上盖图形的顶点
%信息
N=length(yuan); %N 和 N2 分别为 yuan 和 yuan2 的
%像素个数
N2=length(yuan2);
xg=sum(yuan(:,1))/N; yg=sum(yuan(:,2))/N;
zg=sum(yuan(:,3))/N;
xg2=sum(yuan2(:,1))/N2;
yg2=sum(yuan2(:,2))/N2;
zg2=sum(yuan2(:,3))/N2;
%求 yuan 和 yuan2 的质心, 见公式 (2)
T1=[1 0 0 0; 0 1 0 0; 0 0 1 0; -xg -yg -zg 1];
T3=[1 0 0 0; 0 1 0 0; 0 0 1 0; xg yg zg 1];
T12=[1 0 0 0; 0 1 0 0; 0 0 1 0; -xg2 -yg2 -zg2 1];
T32=[1 0 0 0; 0 1 0 0; 0 0 1 0; xg2 yg2 zg2 1];
%3D 旋转矩阵中的 T1 和 T3, 见公式 (2)、(4)
M=24; %动画画面的帧数, M=24
mov=avifile('brainRotate.avi'); %创建脑部旋转动画文件
'brainRotate.avi'
for j=1:M %以循环方式产生、显示与保存脑部旋转动画
xian=0;
xian2=0;
%xian 和 xian2 的初始化。xian 和 xian2 分别为 yuan 和
%yuan2 每次旋转后的动画画面。
th=2*pi/M*j; %每次绕 Z 轴旋转的角度 th。
a=0; %每次绕 X 轴旋转的角度 a。
b=0; %每次绕 Y 轴旋转的角度 b。
Rx=[1 0 0 0; 0 cos(a) sin(a) 0; 0 -sin(a) cos(a) 0; 0
0 0 1];
Ry=[cos(b) 0 -sin(b) 0; 0 1 0 0; sin(b) 0 cos(b) 0;
0 0 0 1];
Rz=[cos(th) sin(th) 0 0; -sin(th) cos(th) 0 0; 0 0 1
0; 0 0 0 1];
T2=Rx*Ry*Rz; %3D 旋转变换 T2, 见公式 (3)
T=T1*T2*T3; %脑部图形及上盖绕自身质心旋转的变
%换矩阵 T 和 TT, 见公式 (5)
TT=T12*T2*T32;
xian=[yuan ones(N,1)]*T; %脑部图形及上盖的 3D
%旋转变换, 见公式 (6)
xian2=[yuan2 ones(N2,1)]*T;
xian=xian(:,1:3); %脑部绕其质心在平行于 z 轴的方向
%旋转 360 度 (共 24 帧)
xian2=xian2(:,1:3); %脑盖上盖绕其质心在平行于
%z 轴的方向旋转 360 度 (共 24 帧)
daspect([1,1,0.4]); view(3)
patch('Vertices',xian, 'Faces',fv.faces, 'Facecolor',[1,0.75,
0.65], 'EdgeColor','none');
hold on;
patch('Vertices',xian2, 'Faces',fv2.faces, 'FaceColor',[1,0.
75,0.65], 'EdgeColor','none');
%分别对脑部及其上盖进行面绘制三维重建。顶点为已经
%过 3D 旋转的 xian 与 xian2, 而表面信息不变, 仍然是
%fv.faces 与 fv2.faces, 并设定图形表面颜色与边缘颜色。
```

(下转到 120 页)

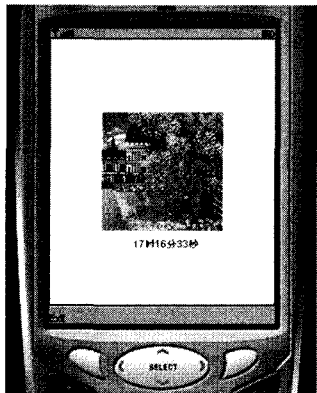


图 2 通过仿真器运行的动画屏保

Saver。通过 SELECT 选中它就可看到运行结果。如图 2 所示。

发布 Java 手机游戏之前，必须要在真实手机上对 MIDlet 套件测试通过。可以通过将应用程序部署到 Web 服务器上，然后使用无线协议技术将该应用程序从服务器传送到手机上；或者将 MIDlet 套件传送到使用蓝牙、红外线或串行连接的手机上。

6 结语

介绍了通过 PngMate 制作手机动画图片的方法，以及如何通过 Timer 类和 TimerTask 类设计完成定时器，并实现了带时钟的动画手机屏幕保护程序。代码在 JDK1.5 和 J2ME Wireless Toolkit 2.2 下调试通过。

(上接第 110 页)

```
hold on;
lightangle (th,30) ;lighting phong; %利用 phone 模型进
%行灯光设置
xlabel ('x') ;ylabel ('y') ;zlabel ('z') ; %显示 X,Y,Z 坐
%标轴
F=getframe; %产生一帧动画
mov=addframe (mov,F) ;
%将动画帧 F 加入动画文件 mov 中
name=strcat ('a',num2str (j)) ;
print ('-tiff',name) ;
%将动画的各帧图片存入名称为 aj.tif 的图像文件中 (j=
%1~24)
if j~=M+1 %擦除上一帧的画面
delete (gca) ;
end
end
aviobj1=close (mov) ; %显示完毕，关闭动画文件
```

5 实验结果

采用上述程序对 Matlab 自带的脑部 MRI 图像序列“MRI” (共 27 幅) 进行三维重建与动画显示，其中旋转参数设置如表 1 所示。

表 1 脑部动画的旋转参数设置

动画的帧数	总的旋转角度	α	β	γ
24	360°	0°	0°	15°

利用 MC 算法的脑部面绘制三维重建结果如图 1 所示，其旋转动画各帧画面如图 2 所示。

利用上述算法进行脑部三维重建及动画生成、显示与保存的总时间为：19.75s，动画文件大小为 126KB 字节。

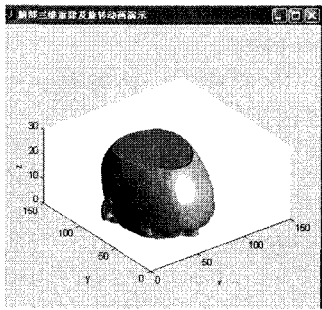


图 1 脑部 MRI 图像的三维重建



图 2 脑部旋转动画的第 1,5,10,15,20 帧画面

6 结语

提出一种三维重建与动画显示算法并采用 Matlab 软件编程实现，实验结果显示该方法能对脑部进行全方位显示，且运行速度较快、占用的内存空间较小。利用该算法并结合图像分割技术可对人体所有器官组织进行三维可视化工作。

参考文献

[1] 田捷, 赵明昌, 何晖光. 集成化医学影像算法平台理论与实践 [M]. 北京: 清华大学出版社, 2004: 21-63.
[2] 成思源, 张群瞻. 计算机图形学 [M]. 北京: 冶金工业出版社, 2003:193-255.
[3] 戴虹. 距下关节 CT 序列图像分割及信息提取研究 [J]. 上海理工大学学报, 2007, 29 (2) :165-169.

(上接第 115 页)

后调用该端点对应的设备驱动程序中的 write () 函数实现。

参考文献

[1] 马尾. 计算机 USB 系统原理及主从机设计 [M]. 北京: 北京航空航天大学出版社, 2004.
[2] 王云飞. USB 系统研究 [M]. 北京: 清华大学, 2001.

[3] STEVE FURBER.ARM SoC structure [M]. 北京: 北京航空航天大学出版社, 2002.
[4] 梁孔科, 杨林楠, 张丽莲. 基于 S3C2410 与 Linux 的 USB 驱动的设计 [J]. 福建电脑, 2008, (4).