

基于OpenGL可编程着色器实现基本图像形态学处理

刘亮 李辉 段康

陕西长岭电子科技有限责任公司 陕西 宝鸡 721006

摘要: 本文主要介绍了基于OpenGL可编程着色器实现基本图像形态学处理腐蚀、膨胀的方法,利用GPU并行计算及硬件加速能力,可满足实时性较高应用场景下的图像处理要求。

关键词: OpenGL, 可编程着色器, 腐蚀, 膨胀

1 概述

常用的图像形态学处理的方法包括腐蚀、膨胀、开运算、闭运算、顶帽运算、底帽运算。腐蚀和膨胀是最基本的处理方法,其它方法都是腐蚀和膨胀方法相互组合产生的。本文主要介绍通过OpenGL可编程着色器实现腐蚀、膨胀处理的方法。

2 腐蚀、膨胀原理

腐蚀和膨胀可看作图像A和结构元素B的卷积运算。结构元素B的可以具有多种结构,如矩形结构、椭圆形结构、十字交叉形结构等,一般采用行列均为奇数的矩阵表示。在腐蚀和膨胀处理过程中,结构元素B的中心原点在图像A上平移,类似卷积的形式重新计算图像A每个点的值。本文中选择的结构元素为矩形结构。

图像A被结构元素B腐蚀,记为 $A \ominus B$ 。结构元素B为值全为1的奇数行列矩阵,结构元素B在图像A上平移,选择结构元素B在图像A上覆盖区域内的最小值,作为图像A在结构元素B中心原点处的腐蚀后结果。

图像A被结构元素B膨胀,记为 $A \oplus B$ 。结构元素B为值全为1的奇数行列矩阵,结构元素B在图像A上平移,选择结构元素B在图像A上覆盖区域内的最大值,作为图像A在结构元素B中心原点处的膨胀后结果。

3 OpenGL 着色器编程

OpenGL是用于渲染2D、3D矢量图形的跨语言、跨平台的应用程序编程接口,用来绘制从简单的图形到复杂的三维景象。OpenGL着色语言(OpenGL Shading Language, GLSL)是用来在OpenGL中着色编程的语言,是一种具有C/C++风格的高级过程语言,着色器是执行在GPU上的程序。着色器至少需要两个,分别为顶点一个着色器(Vertex Shader)和一个片段着色器(Fragment Shader)^[1]。顶点着色器主要用于顶点变换,片段着色器用于片段处理及图像渲染。

着色器编程初始化及绘制步骤如下:

1) 创建着色器程序、顶点着色器和片段着色器;

2) 上传顶点着色器、片段着色器代码并编译着色器代码;

3) 附加顶点着色器、片段着色器到着色器程序,并编译顶点着色器、片段着色器;

4) 链接着色器程序;

5) 创建图像纹理,并上传纹理、设置纹理参数。

示例代码片段如下:

```
glsl = glCreateProgram();
```

```
vshader = glCreateShader(GL_VERTEX_SHADER);
```

```
glShaderSource(vshader,1,&vert,0);
```

```
fshader = glCreateShader(GL_FRAGMENT_SHADER);
```

```
glShaderSource(fshader,1,&frag,0);
```

```
glCompileShader(vshader);
```

```
glAttachShader(glsl,vshader);
```

```
glCompileShader(fshader);
```

```
glAttachShader(glsl,fshader);
```

```
glLinkProgram(glsl);
```

```
samplerLoc = glGetUniformLocation(glsl,"o_texture");
```

```
sizeLoc = glGetUniformLocation(glsl,"size");
```

```
vLoc = glGetAttribLocation(glsl,"a_position");
```

```
fLoc = glGetAttribLocation(glsl,"a_texCoord");
```

```
glGenTextures(1,&tex);
```

```
glActiveTexture(GL_TEXTURE1);
```

```
glBindTexture(GL_TEXTURE_2D,tex);
```

```
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
```

```
glTexImage2D(GL_TEXTURE_2D,0,GL_RGBA,img.  
width(),img.height(),0,GL_RGBA,GL_UNSIGNED_  
BYTE,img.bits());
```

```
glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_
```

```

MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_
MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_
WRAP_S, GL_CLAMP_TO_BORDER);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_
WRAP_T, GL_CLAMP_TO_BORDER);

```

上述代码中初始化着色器程序glsl, 编译顶点着色器vshader和片段着色器fshader, 并链接着色器。创建纹理对象tex, 并将图像img数据上传至纹理对象tex中。

OpenGL在绘制图像时, 启用着色器程序将OpenGL渲染管线切换至着色器模式, 并使用当前着色器进行渲染。着色器根据顶点位置对纹理进行采样, 对采样位置处像素值进行计算, 完成腐蚀和膨胀处理。

4 着色器实现腐蚀、膨胀处理

腐蚀、膨胀主要在片段着色器中完成。以纹理坐标为中心, 根据结构元素B的大小, 计算结构元素B覆盖区域内所有元素的纹理坐标位置, 对这些位置进行纹理采样, 获取该区域内像素值, 计算最大值和最小值, 当腐蚀处理时, 将最小值作为最终输出像素值, 当膨胀处理时, 将最大值作为最终输出像素值。

腐蚀时片段着色器代码示例如下:

```

for (int i = -3; i <= 3; i++) {
for (int j = -3; j <= 3; j++) {
float x = v_texCoord.x + i * delta;
float y = v_texCoord.y + j * delta;
vec4 color = texture2D(o_texture, vec2(x, y));
outColor = min(outColor, color);
}
}

```

上述片段中, 结构元素的行列大小均为7, v_

texCoord为纹理坐标, delta为纹理像素偏移, texture2D为纹理采样函数, o_texture为纹理对象, color为纹理采样像素值。将结构元素B中所覆盖区域内纹理采样, 寻找最小值作为输出像素。

当膨胀时, 将“outColor = min(outColor, color);”替换为“outColor = max(outColor, color);”即可, 其中min函数为计算并返回outColor和color中的较小值, max函数计算并返回outColor和color较大值。

通过着色器实现的腐蚀、膨胀处理效果如图1所示。

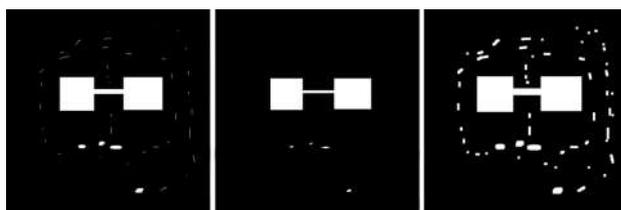


图1 原图(左)、腐蚀(中)、膨胀(右)效果图

如图1所示, 腐蚀将滤除图像噪声或细小部分, 缩小图像, 膨胀将对图像所占区域放大、增强。本文示例中, 图像的宽高尺寸512, 以30帧/秒速度处理图像, 使用OpenGL着色器实现腐蚀或膨胀功能时, CPU(Intel I5-2450M平台下)占用率不足1%, 充分利用GPU并行处理及硬件加速能力。

5 结束语

腐蚀、膨胀等图像形态学处理可以实现图像滤除噪声、图像增强等功能, 腐蚀、膨胀处理的叠加使用, 也可达到不同的图像处理效果。通过OpenGL着色器编程, 可以方便的将图像处理算法在GPU中实现, 有效利用GPU并行计算硬件加速能力, 可应用实时性要求较高的雷达图像显示滤波、增强等场景。

参考文献:

[1][美]Richard S Wright,等.OpenGL超级宝典(第5版)[M].付飞,等,译.北京:人民邮电出版社,2012:164.