**文 献 综 述**



题目：基于WebGL的书法匾额生成

学 院： 信息工程学院

专 业： 计算机科学与技术

班 级： 计算机151

姓 名： 涂渊辉

指导老师： 章夏芬

完成日期： 2019年3月

# 引言

随着电子工具的普及，需要人们亲手写字的情况越来越少，使得很多人都到了提笔忘字的地步。但在日常生活中人们常常会用到书法字，比如设计自己的店铺招牌、签名和匾额等。当需要手写而又不会写时，人们通常请书法家代写或自己在网上找单字合成，但请书法家书写成本颇高，网上也可能找不到合适的字体。本课题就是为了给大家提供方便快捷的书法服务。

为了解决这个问题，本系统设计了一个基于WebGL的书法匾额生成系统，用户通过输入汉字来挑选所需的书法字，系统再对这些书法字进行调整和渲染最终生成用户所需的书法匾额。因为文字图像颜色不一，这样生成的匾额并不美观也不实用。文字图像的大小也不一定相同这样对缩放也会造成不必要的麻烦。因此需先将书法字挑选出来进行矢量化，字体调整好后再利用WebGL进行渲染使生成的匾额更加美观。以上这些即是本课题研究解决的问题，为了解决这些问题，本次课题主要从书法字的查询，矢量化，WebGL渲染，书法匾额设计这些方面去研究，最终实现书法匾额的生成。

书法匾额的设计与生成不仅对文化传承与发扬有重要意义，在现实生活中也可以帮助人们更方便的制作匾额、个性签名，还可以供学生临摹练习书法。本系统对书法字库的设计具有一定的理论价值，为书法字匾额的合成提供了方便，并在一定程度上减少了人们的经济成本。

# 要解决的问题

基于WebGL的书法匾额生成系统可分为书法字的查询、书法字矢量化、书法字渲染、书法匾额合成与3D展示四个主要模块：

1. **生成所需的书法字查询**：在用户输入汉字后，系统连接文字图像数据库筛选出该汉字的不同字体，并输出供用户挑选。
2. **书法字图像的矢量化**：由于文字图像数据库中的文字大小不一致，所以在生成匾额时需要对其进行缩放。但直接进行放大会产生锯齿，使生成的匾额不美观，为了避免这种情况需要先将文字图像矢量化。
3. **基于WebGL的书法字生成及渲染**：书法字矢量化后，通过WebGL使矢量化文字图像更加美观，生成一个个渲染过的单字。
4. **书法匾额合成与3D展示**：利用计算机图形学中三维变换知识，设计一个贴近现实建筑物的立体图形，并为该图形的一个面添加背景。再调整渲染过的单字大小和间距并贴在该面上展示。

# 相关研究

本毕业设计主要通过图像矢量化，WebGL渲染技术[17]来实现书法匾额的生成。矢量化是数字图像处理中的一个重要问题，是一个综合了计算机视觉，计算机图像处理，计算机图形学和人工智能等各个学科的交叉课题。矢量图像有很多优点：(1)矢量图像由简单的几何图元组成,表示紧凑，所占存储空间小；(2)矢量图像易于进行编辑，对矢量图像进行编辑的时候，如进行旋转、拉伸、平移等操作时仅需要修改相应几何图元的参数信息；(3)用矢量表示的对象易于放大或者压缩，而且不会降低其在计算机中的显示质量，矢量图像的放缩能够保持边角的尖锐等特性，不会出现模糊影响显示质量。书法字矢量化主要包括书法字二值化、书法字轮廓提取、书法字轮廓拟合、书法字轮廓填充。

现阶段，图像矢量化的算法主要有拟合算法[1]、动态跟踪算法[2]。在图像处理中有一种常见的处理就是对图像进行轮廓提取和轮廓跟踪。轮廓跟踪的基本方法是：先根据某些严格的探测准则找出目标物体轮廓上的像素，再根据这些像素的某些特征用一定的跟踪准则找出目标物体上的其他像素。

## 书法字二值化

### 3.1.1 书法字灰度化

在RGB模型中，如果R=G=B时，则彩色表示一种灰度颜色，其中R=G=B的值叫灰度值，因此，灰度图像每个像素只需一个字节存放灰度值（又称强度值、亮度值），灰度范围为0-255。一般有分量法、最大值法、平均值法和加权平均法四种方法对彩色图像进行灰度化。

1.分量法

将彩色图像中的三分量的亮度作为三个灰度图像的灰度值，可根据应用需要选取一种灰度图像。

  

其中为转换后的灰度图像在处的灰度值。

2.最大值法

将彩色图像中的三分量亮度的最大值作为灰度图的灰度值。



3.平均值法

将彩色图像中的三分量亮度求平均得到一个灰度值。



4.加权平均法

根据重要性及其它指标，将三个分量以不同的权值进行加权平均。由于人眼对绿色的敏感最高，对蓝色敏感最低，因此，按下式对RGB三分量进行加权平均能得到较合理的灰度图像。



### 3.1.2 阈值计算

阈值选取方法，但如同其他图像分割算法一样，没有一个现有方法对各种各样的图像都能得到令人满意的结果。一般有以下几种方法：

（1）灰度平局值法[3]，即使用整幅图像的灰度平均值作为二值化的阈值，一般该方法可作为其他方法的初始猜想值；

（2）百分比阈值（P-Tile法）[4]，Doyle于1962年提出的P-Tile (即P分位数法)，可以说是最古老的一种阈值选取方法。该方法根据先验概率来设定阈值，使得二值化后的目标或背景像素比例等于先验概率，该方法简单高效，但是对于先验概率难于估计的图像却无能为力；

（3）基于谷底最小值的阈值[5]，此方法实用于具有明显双峰直方图的图像，其寻找双峰的谷底作为阈值，但是该方法不一定能获得阈值，对于那些具有平坦的直方图或单峰图像，该方法不合适；

（4）基于双峰平均值的阈值[6]该算法和基于谷底最小值的阈值方法类似，只是最后一步不是取得双峰之间的谷底值，而是取双峰的平均值作为阈值；

（5）迭代最佳阈值[7]，该算法先假定一个阈值，然后计算在该阈值下的前景和背景的中心值，当前景和背景中心值得平均值和假定的阈值相同时，则迭代中止，并以此值为阈值进行二值化；

（6）OSTU大津法[8]，该算法是1979年由日本大津提出的，该算法是一种自适应的阈值确定的方法，主要是思想是取某个阈值，使得前景和背景两类的类间方差最大。它是按图像的灰度特性,将图像分成[背景](http://zh.wikipedia.org/wiki/%E8%83%8C%E6%99%AF" \o "背景)和目标两部分。

由于OSTU大津法算法被认为是图像分割中阈值选取的最佳算法，计算简单，不受图像亮度和对比度的影响。因此，使类间方差最大的分割意味着错分概率最小。所以本系统拟采用OSTU大律法计算阈值，通过遍历整幅灰度图像，来获取到区别黑白两色的最佳阈值，从而获得只有黑白两色的书法字，简化处理对象。

### 3.1.3 二值化方法

图像二值化[3]（ Image Binarization）就是将图像上的像素点的灰度值设置为0或255，也就是将整个图像呈现出明显的黑白效果的过程，图像的二值化使图像中数据量大为减少，从而能凸显出目标的轮廓。

把灰度图像二值化，得到二值化图像，这样子有利于在对图像做进一步处理时，图像的集合性质只与像素值为0或255的点的位置有关，不再涉及像素的多级值，使处理变得简单，而且数据的处理和压缩量小。为了得到理想的二值图像，一般采用封闭、连通的边界定义不交叠的区域。所有灰度大于或等于阈值的像素被判定为属于特定物体，其灰度值为255表示，否则这些像素点被排除在物体区域以外，灰度值为0，表示背景或者例外的物体区域。

## 3.2书法字轮廓提取

二值图像的轮廓提取对于图像识别，图像分割有着重要意义。该算法的核心就是将图像目标的内部点消除。对于书法字轮廓提取来说，也是基于对图像轮廓提取的研究而来。图像边缘信息主要集中在高频段，通常说图像锐化或检测边缘，实质就是高频滤波。我们知道微分运算是求信号的变化率，具有加强高频分量的作用。在空域运算中来说，对图像的锐化就是计算微分。由于数字图像的离散信号，微分运算就变成计算差分或梯度。图像处理中有多种边缘检测（梯度）算子，常用的包括普通一阶差分，Robert算子（交叉差分）[9]，Sobel算子[9]等，是基于寻找梯度强度。拉普拉斯算子[10]（二阶差分）是基于过零点检测。通过计算梯度，设置阀值，得到边缘图像。在本系统中，由于处理的书法字图像结构较为简单，直接采用基于当前像素点的4-邻域法来消除内部点即可，这样就可以获得多条只有一个像素宽的闭合轮廓。

二值图像轮廓跟踪[11]用于提取二值图像中的目标区域，以便对目标区域做进一步处理，如：区域填 充，计算轮廓长度、面积、重心，特征提取和图像识别等。我们可以采用链码和线段表两种不同的方法来描述轮廓。链码方式描述轮廓需要记录轮廓的起点和轮廓上 每一点相对于前一点的链码值列表。以8-领域链码为例，首先按照从左至右，从上至下的顺序查找轮廓起点；再按照右、右下、下、左下的顺序查找第二个轮廓点；然后按照右、右下、下、左下、左、左上、上、上右的顺序查找其它轮廓点，直到找到轮廓起点为止。

## 3.3书法字轮廓拟合

在边缘检测中总会提取出不连续点，或伪轮廓。在这种情况下需要拟合出目标的轮廓，要采用合适的轮廓拟合算法，拟合出与原曲线最相近的曲线。轮廓拟合的算法有很多种，其中使用Bezier曲线（贝塞尔曲线）[12]来表示字形有较高的精确度以及更灵活的存储及处理方式。Bezier curve又称贝兹曲线或贝济埃曲线，它是由法国工程师皮埃尔·贝塞尔于1962年发现，他运用贝塞尔曲线来为汽车的主体进行设计。贝塞尔曲线最初由Paul de Casteljau于1959年运用de Casteljau演算法开发，以稳定数值的方法求出贝兹曲线。贝塞尔曲线的起始点和终止点为锁定的，然后在中间计算出一些控制点来拟合出一条曲线。

给定点P0、P1，线性贝兹曲线只是一条两点之间的直线。这条线由下式给出：



二次方公式

二次方贝兹曲线的路径由给定点P0、P1、P2的函数B（t）追踪：



三次方公式

P0、P1、P2、P3四个点在平面或在三维空间中定义了三次方贝兹曲线。曲线起始于P0走向P1，并从P2的方向来到P3。P0和P1之间的间距，决定了曲线在转而趋进P3之前，走向P2方向的“长度有多长”。曲线的参数形式为：



## 3.4书法字轮廓填充

书法字轮廓填充其实是一个将指定不规则区域内部像素填充为填充色的过程。填充算法在计算机辅助设计和图像处理等领域有广泛应用。包括了注入填充区域算法[13]、种子填充算法[14]、扫描线填充算法[15]、边填充算法[16]等。

（1）注入填充算法(FloodFill Algorithm)用于内部定义区域，以改变整个区域的颜色属性，它把区域内的原像素点值改变成另一种像素点值。此算法所采用的基本方法是首先确定(x; y)点的像素点是否在区域内尚未被访问过的那一部分之中，也就是说，如果这个像素点的值是原始值old-value，则需要把它改为填充的值new-value，然后按八连通区域性质先后访问其八个相邻的像素点，当访问其中每一个近邻像素点时，都要进行递归调用。此算法通过在四个方向而不是八个方向上扩展，就可以用来填充一个内部定义的四连通式区域。

（2）种子填充算法又称为边界填充算法。其基本思想是：从多边形区域的一个内点开始，由内向外用给定的颜色画点直到边界为止。如果边界是以一种颜色指定的，则种子填充算法可逐个像素地处理直到遇到边界颜色为止。种子填充算法常用四连通域和八连通域技术进行填充操作。

从区域内任意一点出发，通过上、下、左、右四个方向到达区域内的任意像素。用这种方法填充的区域就称为四连通域；这种填充方法称为四向连通算法。

从区域内任意一点出发，通过上、下、左、右、左上、左下、右上和右下八个方向到达区域内的任意像素。用这种方法填充的区域就称为八连通域；这种填充方法称为八向连通算法。一般来说，八向连通算法可以填充四向连通区域，而四向连通算法有时不能填充八向连通区域。

（3）扫描线填充算法的基本思想是：用水平扫描线从上到下（或从下到上）扫描由多条首尾相连的线段构成的多边形，每根扫描线与多边形的某些边产生一系列交点。将这些交点按照x坐标排序，将排序后的点两两成对，作为线段的两个端点，以所填的颜色画水平直线。多边形被扫描完毕后，颜色填充也就完成了。扫描线填充算法也可以归纳为以下4个步骤：

1、求交，计算扫描线与多边形的交点

2、交点排序，对第2步得到的交点按照x值从小到大进行排序；

3、颜色填充，对排序后的交点两两组成一个水平线段，以画线段的方式进行颜色填充；

4、是否完成多边形扫描？如果是就结束算法，如果不是就改变扫描线，然后转第1步继续处理。

（4）边缘填充算法是先求出多边形的每条边与扫描线的交点，然后将交点右侧的所有像素颜色全部取为补色（或反色）。按任意顺序处理完多边形的所有边后，就完成了多边形的填充任务。边缘填充算法利用了图像处理中的求“补”或求“反”的概念，对于黑白图像，求补就是把RGB(1,1,1)（白色）的像素置为RGB(0,0,0)（黑色），反之亦然；对于彩色图像，求补就是将背景色置为填充色，反之亦然。求补的一条基本性质是一个像素求补两次就恢复为原色。如果多边形内部的像素被求补偶数次，保持原色，如果被求补奇数次，显示填充色。

## 3.5 WebGL渲染

WebGL（全写Web Graphics Library）是一种3D绘图协议[17][18]，这种绘图技术标准允许把JavaScript和OpenGL ES 2.0结合在一起，通过增加OpenGL ES 2.0的一个JavaScript绑定，WebGL可以为HTML5 Canvas提供硬件3D加速渲染，这样Web开发人员就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了，还能创建复杂的导航和数据视觉化。显然，WebGL技术标准免去了开发网页专用渲染插件的麻烦。

WebGL定义为一个3D图形库，并以标准、高效的方式支持互联网浏览器的3D场景渲染操作。根据维基百科中所描述的内容，渲染操作可视为源自模型的图像生成过程，并通过计算机程序予以实现。由于该过程可表示为计算机处理行为，因而存在多种方式生成此类图像。差别一在于是否需要使用特定的图形硬件。对于软件渲染，3D场最所需计算均在CPU中执行；相应地，硬件渲染则是指图形处理单元(GPU) 以实时方式执行3D图形计算。从技术视角来看，由于硬件负责某些专属操作，因而与软件渲染相比，其行为更加高效。鉴于独立于硬件支持，因而软件渲染方案应用范围较大。WebGL采用了客户端渲染方案，3D场景素材通过服务器下载，但图像处理过程依赖于客户端的图形硬件并在本地予以执行。WcbGL可视为类3D Web图形API,因而无须执行安装步骤，但应提供以下几种互联网浏览器中的任意一种：Firefox 4.0浏览器(或以上版本)、Google Chrome 11浏览器(或以上版本)、Safari浏览器(OSX 10.6或以上版本)、Opera 12浏览器(或以上版本)。

在本系统中，当一个个书法字矢量化处理好后便可进行基于WebGL的渲染。用于渲染的属性有很多如文字的尺寸、拉伸的深度(也就是高度height)、颜色、透明度、凹凸效果、高光效果、光照效果、纹理，另外，还有一个可选参数用于倾斜文字。

# 研究的方案与内容

## 4.1系统框架

本系统提供书法匾额生成的功能，主要可分为：书法字的查询、书法字图像的矢量化、基于WebGL的书法字生成及渲染、书法匾额合成与3D展示四个基本功能。用户进入系统后输入要查询的书法字，系统会根据用户的输入找出相关的字体对象。在用户挑选字体后，系统自动对原始书法字进行矢量化处理。经过矢量化处理后得到的矢量化文字图像再进行渲染；最后进行坐标变换、匾额合成、三维投影输出。整个系统主要框架如图4.1所示，主要分为书法字挑选、书法字矢量化、书法字渲染和书法匾额设计与展示四个模块。

**书法字矢量化**

书法字二值化

书法字轮廓提取

书法字轮廓拟合

书法字轮廓填充

**基于WebGL的书法字渲染**

输入

用户

输入汉字

选择书法字

**书法字挑选**

**书法字数据库**

原贴图像

矢量化图像

输出

匾额背景设计

匾额形状设计

调整文字大小

调整文字间距

展示

贴上单字

**书法匾额设计与展示**

图4.1 系统主框架流程图

## 4.2拟采用的方案

本课题是实现一个基于WebGL的书法匾额生成系统，本系统实现的关键在于书法字的查询、书法字的矢量化、基于WebGL的书法字渲染、书法匾额的合成上。

1. 书法字的查询

在对书法字进行矢量化之前，首先需要对目标字体进行选择，在这里主要使用ASP从数据库中读取字体（character）表中的字体路径，从而在网页上根据用户请求动态加载字体图像。由于对于相同的一个字体，数据库中往往保存着多张不同风格、不同书法家的字体作品，这时便会访问到多幅字体对象，为了区别不同的字体对象，则需要对每幅字体对象进行横向与纵向位置的编号，例如横向采用动态增长的i值进行编号，纵向采用动态增长的j值进行编号，这样每一幅字体图像将会具有唯一的标识(i,j)。

2.书法字矢量化

书法字矢量化包括书法字二值化、书法字轮廓提取、书法字轮廓拟合、书法字轮廓填充。书法字图像预处理主要包括书法字灰度化和二值化处理。

其中书法字的灰度化一般都是为了减小图像原始数据量，便于减少后续处理计算量综合各方法的性能与处理速度，最终决定采用加权平均值法处理每一个像素点，来简化图像色彩处理。而对于图像二值化，本系统拟采用OSTU大律法计算阈值，通过遍历整幅灰度图像，来获取到区别黑白两色的最佳阈值，从而获得只有黑白两色的书法字，简化处理对象。

对于书法字轮廓提取，由于在本系统中处理的书法字图像结构较为简单，直接采用基于当前像素点的4-邻域法来消除内部点即可，这样就可以获得多条只有一个像素宽的闭合轮廓。假设当前像素P(x,y)的四个邻域像素满足如下条件，则该点即为内部点：（1）P(x,y)为目标像素点，假设目标像素点为黑色0，背景像素为白色255，那么P(x,y)=0；（2）P(x,y)的四个邻域像素均为目标像素0。

对于一条曲线来说，端点控制点和中间两个像素点获取后，便需要利用拟合公式模拟出最接近原始图像的轮廓。本系统采用基于三阶Bezier曲线的拟合方法，通过轮廓点提取出合适的端点控制点，然后再从每两个相邻的端点控制点之间选取另外两个合适的像素点，最后通过三阶Bezier曲线计算出每条曲线的四个控制点，从而实现高度拟合。

对于填充而言，由于扫描线法和种子填充法处理图像时所需要的时间都较长，本系统拟使用<canvas>中自带的闭合区域填充法进行区域填充，该方法首先需要调用beginPath()方法来开始一段新的路径，bezierCurveTo()方法创建一条三次方贝塞尔曲线的路径，然后应用程序在绘图环境对象上调用stroke()与fill()方法，对刚才那些路径进行描边或填充。

3.基于WebGL的书法字渲染

利用WebGL 2.0基于OpenGL ES 3.0，确保了提供许多选择性的WebGL 1.0扩展，并引入新的API。可利用部分Javascript实现自动存储器管理。WebGL 以“分而治之”这方式渲染对象，即复杂的多边形可分解为三角形、直线以及点图元。随后，各几何图元经多个步覆井通过GPU以并行方式加以传递，即渲染管线，进而生成canvas所显示的最终场景。书法字用于渲染的属性有很多如文字的尺寸(世界空间中的单位)、文字拉伸的深度(也就是高度height)、字形的样式、分别用于着色网格前面和侧面的材质(名字分别是material 和extrudeMaterial )，另外，还有一个可选参数用于倾斜文字。

4.书法匾额的合成

书法字渲染过后，就需要进行三维匾额设计，它主要包括三维模型构建和三维匾额合成两个部分。三维模型构建首先需要构建三维立体的基本模型，然后使用正轴侧投影将设计好的三维立体图形投射到二维空间中。本系统拟使用正轴侧投影算法，进行了基本的三维构建，并通过相应的坐标变换，最后使用Html5<canvas>容器进行成品展示。匾额内容检索包括目标字体的检索和纹理背景的检索，先使用ASP对于数据表执行查询操作；然后将查询到的图像路径信息返回给加载界面，该界面将会根据图像的路径向用户呈现出各种各样的纹理背景，用户可以从中选取合适的纹理加载到之前所构建的立体图形的表面；最后，将之前处理好的目标字体贴到三维立体平面上便可以初步生成匾额。

## 参考文献

[1]张立东,基于动态曲线拟合的最大幅度测向算法,空军装备研究院需达与电子对抗研究所,2016年3月。

[2]睢丹，侯德恒,动态图像跟踪优化算法的研究与仿真,安阳师范学院软件学院，河南,2016年6月。

[3]张玲，尹立新，改进的基于平均灰度值特征提取的方法及应用， paper.edu.cn，2009年。

[4]Threshold Segmentation of Road Images Based on the Combination of P-tile and Histogram-Based Fuzzy C-Means Clustering, Computer Engineering & Science ZHANG Yong-jie1,YIN Yuan1,HOU Xiang-dan1,LI Hong-zhen2, OCT 2008.

[5]任会峰，阳春华，周璇，桂卫华，鄢锋， 基于自适应谷底检测的浮选泡沫形态特征提取， 《化工自动化及仪表》 ， 2011年07期。

[6]孙光灵， 基于模糊阈值的图像分割方法研究， 硕士学位论文， 合肥工业大学， 2005年6月。

[7]夏丽， 基于迭代最佳阈值分割的边缘检测研究， 《时代教育》 2012年21期。

[8]卜文斌，游福成，李泉，王惠华，段怀锋，一种基于大津法改进的图像分割方法，北京印刷学院北京，2015年4月。

[9]灰度图像--图像增强 Robert算子、Sobel算子，

https://blog.csdn.net/starzhou/article/details/43375019,访问日期：2019年2月10日。

[10]傅秋桃，拉普拉斯算子的研究，郧阳师范高等专科学校 数学系，湖北，2008年3月。

[11]WQVBJHC的专栏——二值图像轮廓跟踪，

http://blog.csdn.net/wqvbjhc/article/details/6065476,访问日期：2019年2月10日。

[12]贝塞尔曲线原理，https://www.cnblogs.com/hnfxs/p/3148483.html,访问日期：2019年2月10日。

[13]Junsong Zhang, Hongwei Lin, Jinhui Yu，A Novel Method for Vectorizing Historical Documents of Chinese Calligraphy，Zhejiang University，State Key Lab of CAD&CG，Hangzhou, Zhejiang, 310058, China

[14]Orbit专栏——种子填充算法，

http://blog.csdn.net/orbit/article/details/7323090,访问日期：2019年2月10日。

[15]Kai Yula,BinHulb, Milan Ye,Zhenming Yuan,Chinese Calligraphy System in CADAL, School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, China.

[16]边缘填充算法https://www.cnblogs.com/u013533289/p/4995541.html访问日期：2019年2月10日。

[17]Tony Parisi著，郝稼力译，WebGL入门指南，人民邮电出版社，2013年6月.

[18]Diego Cantor Brandon Jones 著，李强译，WebGL编程指南 清华大学出版社，2013年12月。

[19]Andreas Anyuru著，吴文国译，WebGL高级编程——开发Web 3D图形，清华大学出版社，2013年6月。