*位移细分曲线

谢浩[1]，费广正[2]，李欣[1]，韩红雷[1]，赵越[1]
（[1]中国传媒大学计算机与软件学院学院，北京 100024；[2]中国传媒大学动画学院，北京 100024）

摘要：本文给出一种新的曲线表示方法，能在一定压缩比条件下仍然保持原始输入折线的细节特征，并且能为用户提供方便的编辑手段。当用户直接通过 I/O 设备输入或扫描点阵位图中的曲线轮廓作为输入时，输入曲线是一系列离散的点构成的折线段，本文提供简单、自动的构造算法将输入的折线段转换到位移细分曲线。建立位移细分曲线的步骤如下：1. 通过对输入曲线进行预处理以提取特征点；2. 利用曲线细分的方法处理特征折线得到光滑曲线；3. 建立细分曲线与输入曲线的偏移量表；4. 获得由特征折线及位移映射表表示的位移细分曲线。本文曲线表示方法可以应用到人机交互和非真实感图形学中涉及到曲线计算及编辑的领域，其曲线的参数化与细节信息的标量化不仅提高了曲线存储压缩比，最重要的是为曲线提供了精确的描述以及灵活的编辑手段。

关键词：位移映射，细分，控制点，参数化曲线，特征折线

Displaced Subdivision Curve

Xiehao[1]，*Feiguangzheng*[2]，*Linxin*[1]，*Hanhonglei*[1]，*Zhaoyue*[1]

(1 Computer and Software School, Communication University of China, Beijing, 100024)

(2Animation School, Communication University of China, Beijing, 100024)

Abstract: Displaced Subdivision Curve is a new curve representation, paying more attention to preserving details of origin curve as well as maintaining. We present a simple, automatic scheme for converting detailed geometric models into such a representation. First of all, dominant points generally describing the original curve should be extracted from origin curve for preparation. Then we parameterize the dominant points to gain a smooth curve using the curve subdivision algorithm, and calculate displacements in corresponding vertices between the original curve and the subdivided curve taking advantages of the hardware acceleration. Our representation comes from matching displacements to controlling curve. This representation can be used in many fields, such as human-computer interaction，non-photorealistic computer graphics.

Keyword: Displacement mapping, subdivision, control point, parameterized curve, feature polyline

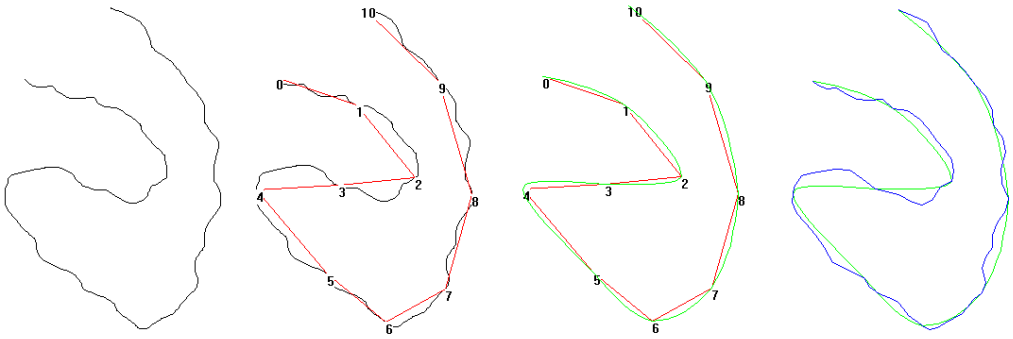

图 1. ①原始输入曲线；②将原始曲线段均匀采样得到的控制折线；
③将简化后的控制线段子分成为光滑曲线；④添加法向量位移后的位移细分曲线。

*本课题受到国家自然科学基金（60403037）资助
本文发表于：中国图像图形学报，2006 年增刊，pp. 113-119

谢浩，1981 年生，硕士研究生，主要研究方向计算机图形学，草图建模和数字娱乐。Email: xiehao@cuc.edu.cn. **费广正**，1973 年生，副教授，主要研究方向为计算机动画、虚拟现实等。吴明峰，1981 年生，硕士研究生，主要研究方向为计算机图形学，人脸建模。**李欣**，1979 年生，硕士研究生，主要研究方向为交互技术。**韩红雷**，1980 年生，硕士研究生，主要研究方向为计算机动画。**赵越**，1981 年生，硕士研究生，主要方向非真实感图形学。

0 引　言

在曲线造型中, 曲线的表示通常通过参数形式、隐式形式或离散化方法(如割角、补角等) 来表达, 其中曲线的参数形式表示使用最广泛。细分、拟合等曲线参数化技术已广泛的应用到计算机辅助设计、真实感图形学和科学可视化等领域，其相关的造型理论与算法已相当丰富。但近年兴起的人机交互技术、非真实感图形学对曲线在可控制性、智能化、个性化及存储方式等方面提出了更高的要求。传统的非真实感技术通常都是采用在参数化曲线上添加噪声来模拟手绘效果，在曲线上不能保留各个艺术家的笔迹、风格等特点。在新一代的人机交互系统中，更为自然的输入方式如手写板等，正逐步走入人们的视野。不管是鼠标、手写板还是智能笔，这些 I/O 设备所采集到的都是离散的非参数化信息，均需要进行数据的参数化、结构化处理，以便分析和存储和修改。传统的曲线表示法采用了曲线拟合或细分对离散折线参数化，以获取可控制性和一定的存储压缩比，但在非真实感图形学技术中的画笔模拟技术中，为了反映艺术家的艺术气质与创造力，风格与个性还需要保存曲线的某些笔触以及细节特征。下面提出的位移细分曲线不仅能参数化曲线提供简单便利的编辑交互，更重要的是它保留了输入曲线的细节信息，保留有艺术家的笔迹、风格。

1 相关工作

以下我们从特征点的提取、曲线细分算法和曲面的位移细分表示等方面概述相关工作：

通常用户通过鼠标或智能笔输入的离散点数目都较大。如果我们直接使用原始数据求取控制曲线或位移序列，曲线控制点将过于密集，对于像素显示设备而言不仅没有实际意义，还会增加存储量。原始数据的另一个缺陷是缺乏结构或拓扑的信息，点集中元素无权重可言，存在不少噪声点。因此我们需要确定能反映曲线主要几何信息的特征点（Dominant Points），参照[8]与[12]的特征点提取方法得出了本文提取特征折线的算法。

过去的几十年中，研究人员提出了很多成熟的曲线细分算法，如早期的 Chaikin[5]提出的角切割思想的生成曲线细分算法，de Casteljau 算法[2]，de Boor 算法[6]、Dyn 等人提出的四点插值细分算法[7]和 Hermite 细分算法[8]。但在传统参数表示方法往往直接利用细分结果表示曲线，使得人机交互中的一些重要细节可能丢失。而在本文的方法中，细分算法被用于提供控制曲线，最终表示由控制曲线和标量位移组成，能有效地保持原始曲线细节特征。本文我们采用最简单的带权相加的插值细分方法，当然采用其他一些参数曲线和细化算法也能得到不错的效果，如 Hermite 曲线及 Hermite 细分算法[8]。

Schröder 和 Sweldens[11]在描述地形时提出了由平面及其上的高度场来表示地形的表示法。之后 Aaron Lee 等人在[1]中进一步提出新的三维模型表达法，即通过控制网格和标量位移场相结合来表达三维几何模型。该算法不仅在视觉效果方面较好的还原了原始三维模型，还获得极高的存储压缩比和良好的特性。

[1][11]中曲面或地形新表示法的给我们的思路以较大的启示。本文正是采用了类似位移细分曲面的标量位移细节和控制参数曲线的方法表示曲线。但是位移细分曲线并不是位移细分曲面[1]的平面简化，[1]中位移细分曲面的特点是对 3D 模型的压缩存储与再现，而位移细分曲线的出发点是在保存输入曲线的细节特点的同时对用户提供简单便利的交互手段。相比之下，文章更重视特征点采样方法，最大限度的提取了输入曲线的形状特征信息。比较[13]中的曲线构造方法，本文的方法无法像 multiresolution curves 一样精确的表达曲线细节，却实现了曲线的压缩。

2 建立位移细分曲线

曲线的输入途径通常有两种形式：1.直接通过鼠标、手写板等 I/O 设备；2.通过读取扫描图片、点阵位图里面的曲线轮廓来获取。在手绘输入设备中，读取的数据都是一系列离散的点，连接起来所形成的输入曲线；在通过点阵位图输入方面，读取的曲线数据是一系列离散像素坐标。如何将这些离散数据信息来建立位移细分曲线，可以分为以下几个步骤：1.对原始离散点采样特征点，得到特征折线；2.对特征折线进行细分得到细分曲线；3.计算子细分曲线各顶点的法向量，并计算细分曲线的每个顶点在法向量方向上与原始曲线的位置偏差；4.保存特征折线与法向量位移表构成位移细分曲线。（图 1）

2.1 采样输入曲线特征点

无论是设备输入，还是位图读取，原始曲线的数据都是一系列有序的点坐标构成，记录了曲线从起点到终点的所有点坐标，该数据是庞大的。这里首先需要对曲线进行简化，也就是对曲线上的点进行采样。在原始曲线上取有限个特征点，连接起来形成特征折线，并且需要让特征折线在一定程度上保持输入曲线的形状。由特征点组成的特征折线是位移细分曲线的存储对象，也是对用户提供修改曲线操作的控制对象。采样的特征点越多，特征曲线越贴近原始曲线；采样点越少，在交互过程中用户控制越方便，存储量越小。

对于特征点的采样方法有以下几种，可以根据不同的情况使用不同的方法：

2.1.1.均匀采样。

均匀采样就是等间隔采样特征点。均匀采样中，采样点个数决定了特征折线与原始曲线的逼近程度。通常，在均匀采样

中，需要按照原始曲线轮廓的复杂程度来决定采样点的个数：如果原始曲线近似于直线，只需要采样起始点与终点；当原始曲线很复杂，则需要成倍的增加均匀采样点数量以保持特征折线逼近输入曲线。由于曲线形状的复杂程度是计算机无法直接获取的信息，在均匀采样中只能根据输入点的数量来估计需要采样的特征点数量或者实行用户交互式的输入特征点数量。要取得理想的逼近特征曲线，均匀采样点的特征点数量通常比较多，这对以后提供用户交互造成了不便。

2.1.2.交互式输入特征点。

这种采样方法是让用户直接指定特征点，除了起点和终点以外，用户必须自己指定输入曲线上所有的特征点。这种特征点提取方法可以让用户自己指定需要控制的特征折线的形状，更方便对其进行修改。缺点是用户必须指定所有的特征点，对使用造成不必要的麻烦。

2.1.3. 全局误差最小化采样。

这种方法可以自动决定采样点数量，自动采样特征点，并且采样后的误差是全局最小化的。这个方法是由 Michael Plass and Maureen Stone 提取曲线特征点算法[12]改变得来。算法中，我们需要引入一个 τ 值来代表多选择一个特征点的代价。τ 值的大小决定采样的特征折线与输入曲线的逼近程度： τ 越大，误差越大，采样点数量越少； τ 越小，误差越小，采样点数量越多；当 τ =0 的时候，该采样算法将会采样所有的输入曲线上的点。对于 τ 的值，可以由程序员经验给出，也可以动态的让用户自己调节。

通常，如果需要在 N 个点中采样 X 个点，一共有 C_n^x 种采样方案，需要在这些方案中选择一个误差最小的采样方案。这种方法的计算量是 O（N!），因输入曲线的信息量通常很大，采用 O（N!）的算法是无法计算的，我们引入 τ 来决定误差等级后，通过指定 τ 的大小可以让程序自动获取采样点的数目，这种算法能让计算量从 O（N!）降到 O（N^2）适合于用来处理输入曲线。

算法中采用一个结合距离误差 error 与花费 τ 的权衡值来决定如何采样特征点，将该权衡值命名为大写 $ERROR_{ij}$ 用来表示输入曲线中第 i 个点到第 j 个点这一段曲线的最好采样方案的权衡值。在每加入一个采样点作为特征点时，我们让权衡值多加上一个 τ ，最后选择 ERROR 最小的方案来决定特征点的采样。error 的值为输入曲线所有点到采样折线的距离误差，该误差可以指定为点到特征折线距离的平方和； τ 为多引入一个特征点时的“代价”。

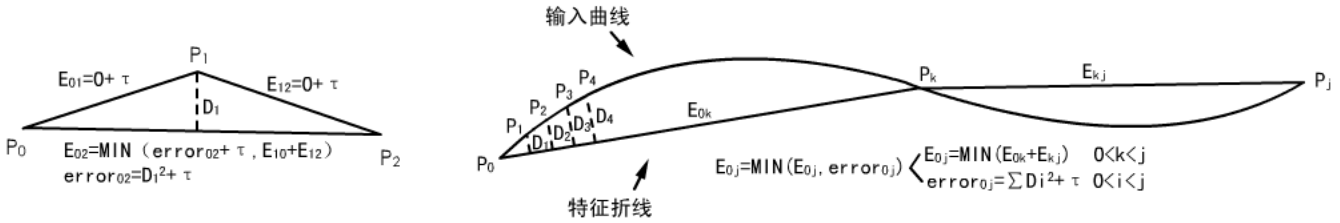

图 2 三个点的最小化 ERROR 方案选择（左图），j+1 个点的最小化 ERROR 方案选择（右图）

如果在 ij 当中插入一个点 k 时，采用

$$ERROR_{ij} = \mathrm{MIN}(ERROR_{ik} + ERROR_{kj}),\ i < k < j。$$

如果都从第一个点开始计算，那么 i 始终为 0。并且因为我们已经存储了 $ERROR_{0k}$ 的值，这里只需要计算 $ERROR_{kj}$ 即可得到 $ERROR_{0j}$。计算所有当 k 从 1 到 j-1 时的 $ERROR_{0j}$，最后得到一个最小 ERROR 作为当前点 j 点的 $ERROR_{0j}$，并且把 k 点保存下来作为最优采样时，经过的采样点。

程序流程如下：

1). 定义一个 N*N 的二维指针数组 KNOTS[n][n]，数组的每个元素需要指向一个 ERROR 结构体。ERROR 结构体包含以下两个元素：

(1)ERROR 存储最小权衡值方案的 ERROR 值。

(2)PassBy 存储采用最小化 ERROR 所经过的前面一个最近的顶点 K。

2). 建立该 N*N 表格，利用下面的函数进行递归。

int GetError (输入点 i，输入点 j){ // 该函数返回 P_i 与 P_j 这段曲线最小误差方案的 ERROR 值

如果 P_i 与 P_j 是同一个点，直接返回 0。

查询 KNOTS 表，如果对应 KNOTS[i][j]已经存在，则直接返回 KNOTS[i][j]保存的 ERROR 值

如果 KNOTS[i][j]方案不存在，则定义一个新 KNOT

先让当前 ERROR 等于曲线在 P_iP_j 间的所有点到直线 P_iP_j 距离的平方和加上 τ

for (k 点从 i 到 j){

递归的计算 $ERROR_{ik}$ 和 $ERROR_{kj}$ 的和，并保存成 k 点的 ERROR 值

if （当前的 ERROR 大于 k 点的 ERROR）{

替换 ERROR 值，并且记录 k 点作为经过点

} } //endfor

在 KNOTS[i][j]中纪录最后得到的 ERROR 值，以及 k 点

返回 ERROR 值 } //函数结束

3). 通过查询建立完的表格，可以取出最小 ERROR 的特征点。该取出算法同样也是一个递归过程。

void SetQueue（Queue *queue, int i）{ //取出 $ERROR_{0i}$ 的接点加入队列 queue 中

如果 KNOTS[0][i]中存储的 PassBy 为 0，则直接返回；

如果 PassBy 不为 0，则将 PassBy 加入特征点队列表；

递归的调用 SetQueue（queue, PassBy）查询的 0 到 PassBy 间是否还有特征点,并继续加入队列}

2.2 对特征折线进行细分

这一步的目的是将已经提取的特征折线进行细分处理，生成光滑细分曲线，并且更逼近于原始曲线。

文章主要对切角法和插值法两种基本细分思想进行比较。需要进行细分的特征折线段是原始输入折线段的采样。在插值细分法中，顶点的位置不改变，也就说细分后曲线的基点仍然是原始输入折线上的点，这样可使得细分曲线在形状上更逼近于原始输入曲线。切角法虽然可以将控制折线段细分成为很光滑的曲线，但是由于我们需要记录的是输入曲线的非真实感细节，对于曲线的光滑度并没有很大的需求。比较切角法和插值法两种细分方法，插值法更利于应用在位移细分曲线算法中。为了简化计算，这里采用对邻近顶点带权相加的方法插入新顶点，权值表可以采取（-1,9,9,-1）/16,这样可以通过位移操作避免进行除法运算。当然如果采用 Hermite 等其他细分方法同样可以得到效果很好的逼近曲线。

确定了使用插值法细分，进一步需要决定细分的次数。细分的次数既决定了曲线的光滑程度，还决定了最终曲线对输入曲线的逼近程度。通常，我们让细分后曲线的顶点数目近似等于原始曲线的顶点数目，这样能最大程度的逼近原始曲线，并且有效的节约存储空间。如果不需要很精确的逼近原始数据，也可以减少细分次数来获得简化曲线，节省存储空间。当原始顶点数为 n_1,特征折线顶点数为 n_2 的时候，细分次数 k 可以如下的公式计算：$k=\log_2(n_1/n_2)$。这样，通过 k 次细分，细分后顶点数接近输入曲线的顶点数。

2.3 建立位移映射表

位移映射是指曲线经过简化再细分后相对于原始输入曲线形成的偏移，只要记录了这些位移量，我们就能保存输入曲线上的细节信息。位移映射表也是位移细分曲线的存储对象。需要计算偏移的对象是细分后曲线上的顶点，偏移的目标是法向量与输入曲线的交点，偏移的方向是顶点的法向量方向。计算各点的偏移量，首先需要计算细分曲线上各点的法向量，并且对法向量单位化；然后求出输入曲线与该法向量直线的交点，最后获取偏移量。

为了减少计算量，并且避免与无关线段求交。在求交点的时候不采取对整个输入曲线求交点，而采用分段求交的方法。因为特征线段上的点为输入曲线的采样点，并且在插值细分后点的位置不改变，由此可以确定细分曲线上的每个点应该对应于输入曲线的某一段。通过遍历相应的曲线段来获取交点，最后求取位移量的方法就是分段求交法。

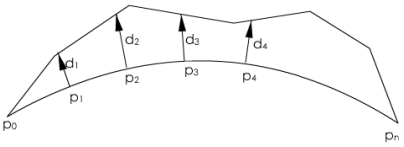

图 3 计算细分曲线上的位移量

2.4 储存与绘制位移细分曲线

在曲线的存储中，只需要对下面两项数据进行存储。

- 特征折线段
- 位移映射表

在占用存储空间方面，特征折线段相对于输入曲线来说只需要占据极小存储空间。位移映射表的记录中，这里只记录每个点的偏移量，对于偏移的方向可以通过计算子分后曲线的法向量进行还原。所以存储空间由输入曲线中的 2 维坐标信息降低到了 1 维偏移量信息。

还原（绘制）位移细分曲线的过程不需要和建立位移细分曲线一样复杂。还原位移细分曲线的步骤如下：1.将存储的特征折线进行细分；2.计算细分曲线顶点的法向量，并且单位化；3.加载法向量位移。

绘制曲线计算量分析：1.细分曲线——插值细分法的计算是通过对附近顶点进行带权值相加来计算插入点位置，采取(-1,9,9,-1)/16 的权值表时，完全可以通过位移运算来避免除法运算。2.法向量计算——法向量方向是通过对该点前后的顶点进行加减运算得到，但是法向量进行单位化时必须进行乘法，甚至开平方运算。3.加载法向量位移——将位移量乘上法向量分量后加载到细分顶点坐标上。综上，还原位移细分曲线的计算量级别是 O(N)。

3 对位移细分曲线的编辑应用

位移细分曲线可以看作是光滑细分曲线与位移贴图相加所组成，特征折线决定的是该光滑曲线，位移映射表则决定了位移贴图信息。通过对特征折线进行编辑，可以实现平移、旋转、缩放、形变等效果；通过对法向量位移表的使用可以修改曲线风格，制作曲线特效。

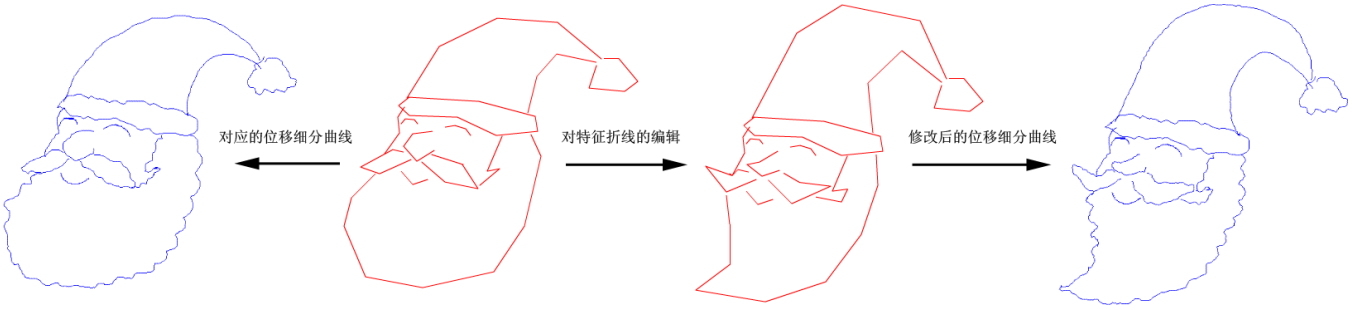

图 4 通过对特征折线的编辑即可得到对应的位移细分曲线，并且保持输入曲线的绘图风格

3.1 平移、旋转、缩放变换以及形状变换

对于平移变换，只需要平移特征折线，对平移后的特征折线重新细分，最后加载位移映射表即可。平移变换不必重新计算法向量以及位移映射表。旋转变换，先对特征折线进行旋转，并且对特征折线细分后重新计算各顶点法向量。旋转变换的法向量计算，可以在原法向量方向上乘以旋转矩阵即可。缩放变换，先对特征折线段进行缩放，细分后不需要进行法向量更新，但为了保持曲线形状不会因为法向量的恢复而走样，需要对位移映射表也进行相应的比例缩放。如果需要的变换是结合了平移、旋转、缩放的平面变换，同样可以推导出如下的变换方法：先对特征折线乘以变换矩阵，将法向量乘以变换矩阵中的旋转变换矩阵，位移映射表乘上缩放比例，之后对特征折线进行细分，加载法向量位移即可。

通过对特征折线段上的顶点位置的编辑可以实现对位移细分曲线的形状修改。将修改后的特征折线段，进一步对特征折线段进行细分、重新计算法向量，最后加载位移映射表恢复细节信息。同样，为了使编辑后的曲线不产生因添加法向量位移导致的走样，我们同样需要按照比例来缩放法向量位移表。缩放的对象是当特征点之间的距离改变后，与之对应的细分曲线顶点的位移表；缩放的比例为修改后的两相邻特征点之间的距离比修改前距离。

3.2 修改线条风格

除了保留输入曲线的风格，用户还可以通过修改法向量位移表来修改曲线的风格。通过在法向量位移表中加入噪音，比如添加了 Perlin Noise 后可以实现曲线的非真实感效果。添加周期函数可以得到锯齿形、波浪形折线。通过对位移映射表的增强，还可以继续添加笔画粗细，笔画风格等信息。

3.3 任意分辨率

对任特征折线段进行任意次细分都可以得到不同的分辨率。细分后按照与其对应的位移曲线来添加法向量位移即可完成任意分辨率。如果需要比存储分辨率还要精细的曲线，则将特征折线不断的细分下去，法向量的偏移取左右两点偏移量的平均值即可。

4 结束语

本文的位移细分曲线利用标量位移保存曲线参数化过程中丢失的细节信息。相对普通参数曲线表示法，本文算法仅额外增加一个标量位移数组，但是抛弃了庞大了二维坐标数组，无论是在表达的准确性、曲线的控制性，还是存储开销方面表现都较为出色，适应新一代人机交互技术和非真实感图形学画笔技术对个性化的要求。而且，由于该表示方法具有多层次的特点，可以更灵活的应用在不同需求的场合。

但是本文构造方法在表现曲线细节方面还存在一些不足有待改进，对一些细节特征复杂的曲线还是无法表现。

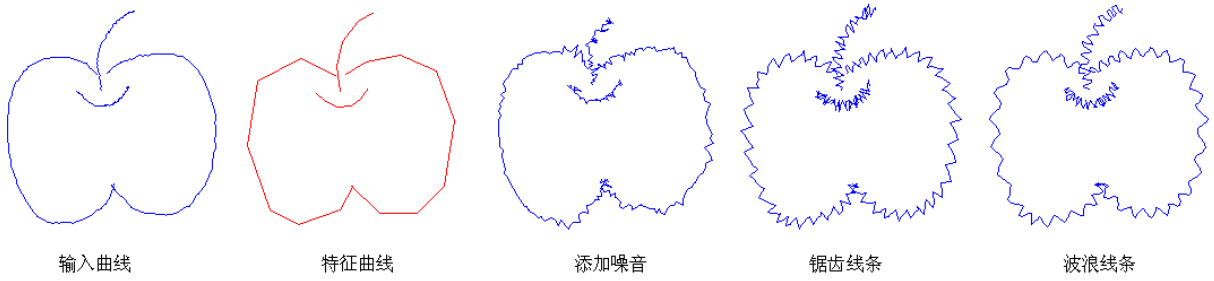

图 5 通过修改位移向量表产生多种线条风格

参考文献

[1] Aaron Lee, Henry Moreton, Hugues Hoppe. Displaced Subdivision Surfaces[A]. Siggraph 2000, Computer Graphics Proceedings[C], 85-94.

[2] Carstensen C, Mühlbach G, Schmidt G. De Casteljau’s algorithm is an extrapolation method [J], Computer Aided Geometric Design, 1995, 12 (4): 371～380.

[3] Tomas Akenine-Möller and Eric Haines. Eric Haines. Real-Time Rendering[M], Second Edition. A K Peters, Ltd, 2002: 270-277.

[4]Randima Fernando, Mark J.Kilgard. The Cg Tutorial The Definitive Guide to Programmable Real-Time Graphics[M]. NVIDIA Corporation, 2003.

[5] R. F. Riesenfeld. An algorithm for high speed curve generation[A]. Computer Graphics and Image Processing[C], 1974, 3(4):346-349.

[6] de Boor C. Cutting corners always works [J], Computer Aided Geometric Design, 1987, 4 (122): 125～131.

[7] Dyn N, Levin D, Gregory J A. A 4-point interpolatory subdivision scheme for curve design [J]. Computer Aided Geometric Design , 1987 , 4 (4) : 257～268.

[8]CHO-HUAK THE, ROLAND T CHIN. On the Detection of Dominant Points on Digital Curves [J]. Trans on Pattern Analysis and Machine Intelligence, 1989, 11(8).

[9]倪明田，吴良芝。计算机图形学[M]。北京大学出版社，1999。

[10]胡云飞，秦严严，戴国忠。基于曲线拟合的笔迹存储和绘制方法研究[J]。计算机工程与科学，2003 年第 25 卷第 5 期。

[11] Schröder, P., and Sweldens, W. Spherical wavelets: efficiently representing functions on the sphere[A]. Proceedings of SIGGRAPH 95,Computer Graphics, Annual Conference Series[C], pp. 161-172.

[12] Michael Plass and Maureen Stone，Curve-Fitting with Piecewise Parametric Cubics[A]，in Proc.SIGGRAPH 83[C], Detroit, July 25-29,1983, pp. 229-236.

[13] Adam Finkelstein and David Salesin , multiresolution curves[A] SIGGRAPH ’1994, Computer Graphics [C], 261-268.