

基于改进稀疏场算法的水平集形状过渡*

翁建广⁺, 庄越挺, 潘云鹤

(浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

Morphing Based on Mended Sparse-Field Algorithm of Level-Set Method

WENG Jian-Guang⁺, ZHUANG Yue-Ting, PAN Yun-He

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: Phn: +86-571-87951853, Fax: +86-571-87951904, E-mail: wengjg@zju.edu.cn

Weng JG, Zhuang YT, Pan YH. Morphing based on mended sparse-field algorithm of level-set method. *Journal of Software*, 2006,17(7):1544-1552. <http://www.jos.org.cn/1000-9825/17/1544.htm>

Abstract: Level-Set method is a good way to do metamorphosis. Narrowband and sparse-field algorithms improve its performance. Results of narrowband morphing are smoother when the sparse-field algorithm is faster. Sparse-field algorithm is mended to fit Euclidean distance model and the narrowband algorithm is used to make up the error of the sparse-field morphing. Topological relationship replaces distance band to define layer sets, and a single side active set is proposed to improve efficiency and robustness. To make up the error of the sparse-field algorithm, which causes obvious alias at the last half stage, two remedy methods are proposed. Averaging and translation method is simpler and more efficient. Narrowband evolution and back method is better for reserving the sharp shape.

Key words: morphing; level-set method; sparse-field; antialias

摘要: 水平集进化是基于体模型进行三维形状过渡的常用方法,窄带算法和稀疏场算法能高效实现水平集进化,窄带算法的结果较为平滑,稀疏场算法速度更快.一方面通过改进稀疏场算法应用于欧氏距离模型提高速度,另一方面运用窄带算法弥补稀疏场算法的误差.提出用拓扑关系代替距离值范围定义各层体素集,并通过单侧活动集定义使算法更为高效和鲁棒.稀疏场算法因为欧氏距离的近似计算引起误差,在过渡的中后期走样明显,为此,提出了均值平移和窄带回退两种反走样方法对过渡模型进行平滑,前者简单、快速,后者失真度低.

关键词: 形状过渡;水平集;稀疏场;反走样

中图法分类号: TP391 文献标识码: A

形状过渡(morphing)是计算机动画中常用的方法,由于能产生新颖的中间形状,可以用于产品设计的外形创新,也是综合推理思想^[1]的一种体现.常用的过渡方法包括二维多边形过渡、二维图像过渡、三维多面体过渡、

* Supported by the National Natural Science Foundation of China under Grant Nos.60533090, 60525108, 60503068 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312100 (国家重点基础研究发展规划(973)); the Natural Science Foundation of Zhejiang Province of China under Grant No.ZD0212 (浙江省自然科学基金); the Science and Technology Project of Zhejiang Province under Grant Nos. 2005C13032, 2005C11001-05 (浙江省科技计划项目重大科技攻关项目)

Received 2004-12-08; Accepted 2005-03-11

三维体模型过渡,其中三维形状过渡的理论和方法较为复杂,其产生的模型用途更为广泛,是目前研究的重点.Kent 等人提出基于多面体模型的过渡^[2]通常要求过渡计算两端的模型必须满足星形的拓扑要求,首先指定两个模型间若干顶点的对应关系,而后进行多面体的拓扑合并,建立所有顶点的对应关系,最后对这些顶点进行插值,实现形状过渡.这种方法对模型的拓扑要求过高,只适用于简单形状.基于体模型的过渡能够实现较为复杂的形状间的过渡,不必受到拓扑结构的约束.体模型形状过渡可以分为特征对应和边界追踪两类.基于特征对应的体过渡由 Leries 等人^[3]提出,他们把二维图像方法推广到三维,首先在源体 S 和目标体 T 指定对应特征,根据对应关系进行空间变化,达到几何对齐,而后根据体素值进行插值,形成新的形状.边界追踪算法早期采用在体素内安置标记点对这些标记点进行追踪,一方面安置标记点的工作量太大,另一方面,在遇到拓扑变化时,标记点方法变得不够稳定.接着出现 VOF 方法,在每个体素内保存体积比函数 C 的值,但体积比函数的求解相当困难.目前较为理想的方法是利用水平集函数 $\phi(x,t)$ 代替 VOF 方法中的体积比函数 C,让水平集函数以曲率相关的速度移动^[4],其零等值面就是模型边界.

水平集进化实现的基本方法是全局算法,每次迭代对所有物体空间内的体素进行计算,效率很低.为此提出了窄带算法^[5]和稀疏场算法^[6],前者把每次进化参与计算的体素局限在等值面周围的一定距离内,后者在街区(city block)距离简化手段的帮助下,把窄带发挥到了极致,效率更高.在边界曲面延伸方向单一的情况下(如向内或向外),Sethian 提出了基于 Eikonal 方程的快进(fast marching)算法^[7],这是水平集方法的一种简化,计算效率较高.基于以上方法,Breen 和 Whitaker 实现了一个三维形状的过渡和编辑系统^[8,9].由于体模型数据量比较大,其稀疏表示也是重要的研究内容^[10],而稀疏场算法和稀疏表示有内在的共通之处,也体现了稀疏场算法的应用前景.

Whitaker 提出稀疏场算法中,根据体素中心点到零等值面的距离定义活动体素集和各外层体素集,并给出了活动体素集的初始化方法.但其还存在以下几个问题: 活动体素集定义基于理想距离模型,在实际进化过程中为保持鲁棒性会损失有效性; 外层体素集根据距离值范围定义,与计算中的拓扑要求不一致; 稀疏场算法的误差引起中后期的过渡中间模型走样明显.

本文第 1 节介绍水平集方法理论和实现,重点分析稀疏场算法的可改进之处.第 2 节给出稀疏场中体素集的重新定义及其相应进化算法.第 3 节描述两种反走样算法,包括均值平移和窄带回退.第 4 节提供实现的技术路线和实例.

1 水平集方法

1.1 理论基础

考虑模型及其周边的三维空间 $\Omega \in R^3$, Ω 封闭有限,其边界是 $S = \partial\Omega$,在三维空间上定义欧氏距离函数 γ :

$$\gamma(x) = d \begin{cases} d = 0, & \forall x \in S \\ d > 0, & \forall x \in \Omega \\ d < 0, & \text{otherwise} \end{cases} \quad (1)$$

等值面的点集合定义为

$$S = \{x \mid \phi(x) = c\} \quad (2)$$

对于边界曲面, c 的值为 0,等值面随时间变化的定义为

$$S_t = \{x \mid \phi(x,t) = c\} \quad (3)$$

设 $s(t)$ 是 S 上的一点,由于 $\phi(x)$ 的值在不同时间保持常量,根据链式法则可以推导得到:

$$\frac{\partial \phi(s(t), t)}{\partial t} + \nabla \phi(s(t), t) \cdot \frac{ds(t)}{dt} = 0 \quad (4)$$

这样,点的运动和距离函数值之间建立了关系,重写该方程得到:

$$\frac{\partial \phi(s(t), t)}{\partial t} = -|\nabla \phi| \frac{ds(t)}{dt} \cdot N(s) \quad (5)$$

设目标模型为 D ,渐变模型和 D 的重叠空间量为

$$M_{\Omega_D, \Omega_i} = \int_{\Omega_i} \gamma_D(x) dx \tag{6}$$

通过把该度量函数表示为边界曲面增量方式,得到关于 Ω_i 的一阶变分:

$$\int_{\Omega_i + d\Omega_i} \gamma(x) dx = \int_{\Omega_i} \gamma_D(x) dx + \int_{S_i} \gamma_D(x) \varepsilon(x) \cdot N(x) dx \tag{7}$$

对 ε 的微分得

$$dM = \gamma_D(x) N(x) \tag{8}$$

基于式(8)的一阶变分和登山法得到曲面边界上每个点的运动方程:

$$\frac{ds(t)}{dt} = \gamma_D(s(t)) N(s(t)) \tag{9}$$

把式(9)代入式(5)并推广到空间范围内所有等值面:

$$\frac{\partial \phi(x, t)}{\partial t} = |\nabla \phi| \gamma_D(x) \tag{10}$$

式(9)和式(10)描述了模型边界曲面的进化过程,详细推导过程请参考文献[4].

1.2 数值算法

体模型把空间划分为体素,不妨设体素是单位大小正立方体,文献[4]提出了在体素空间上用有限差分求解上一节的偏微分方程的方法.设 u^n 是在每个网格中心点对距离函数 $\phi(x, t)$ 的近似值,其中 n 表示经过 n 次进化,在三维空间中, u 值的更新方程为

$$u_{i,j,k}^{n+1} = u_{i,j,k}^n + \Delta t \Delta u_{i,j,k}^n \tag{11}$$

用逆风格式求解 $\Delta u_{i,j,k}^n$,首先记偏导数为

$$\delta_x^+ u_{i,j,k}^n = u_{i+1,j,k}^n - u_{i,j,k}^n, \quad \delta_x^- u_{i,j,k}^n = u_{i,j,k}^n - u_{i-1,j,k}^n \tag{12}$$

Y 和 Z 轴方向的偏导数定义类似.得到 $\Delta u_{i,j,k}^n$ 的求解公式:

$$\Delta u_{i,j,k} = \gamma_D(i, j, k) \cdot \begin{cases} \left(\sum_{w \in x, y, z} \min(\delta_w^+ u_{i,j,k}^n, 0)^2 + \sum_{w \in x, y, z} \max(\delta_w^- u_{i,j,k}^n, 0)^2 \right)^{\frac{1}{2}}, & \text{for } \gamma_D(i, j, k) < 0 \\ \left(\sum_{w \in x, y, z} \max(\delta_w^+ u_{i,j,k}^n, 0)^2 + \sum_{w \in x, y, z} \min(\delta_w^- u_{i,j,k}^n, 0)^2 \right)^{\frac{1}{2}}, & \text{for } \gamma_D(i, j, k) \geq 0 \end{cases} \tag{13}$$

为保持进化的稳定性,每次进化的时间跨度必须满足如下条件:

$$\Delta t \leq \frac{1}{3 \max_{i,j,k} |\gamma_D(i, j, k)|} \tag{14}$$

可以简单地在每次进化中计算所有体素的 u 值,但是这样计算量很大,所以文献[5]提出了窄代算法,提高了计算效率.在文献[6]中,Whitaker 在利用街区距离的计算简单性中提出了稀疏场算法,一方面缩小窄带厚度,另一方面提高了 γ_D 的计算精度,该方法因为这两方面的优势在文献[8,9]所介绍的系统中得以应用.

文献[6]中对稀疏场活动集定义如下:

$$L_0 = \left\{ x \mid -\frac{1}{2} \leq u(x) \leq \frac{1}{2} \right\} \tag{15}$$

外层体素集定义如下:

$$L_n = \begin{cases} \left(x \mid n - \frac{1}{2} < u(x) \leq n + \frac{1}{2} \right), & \text{for } n > 0 \\ \left(x \mid n + \frac{1}{2} > u(x) \geq n - \frac{1}{2} \right), & \text{for } n < 0 \end{cases} \tag{16}$$

其中 n 是整数.但是在三维欧式距离模型中,该定义基于理论距离环境,在实际计算中是不严格的,有一定的冗余.在欧氏三维空间,实现拓扑6分,必须对曲面进行超覆盖^[11],式(17)的活动集定义在实际计算中才比较可靠,

但该活动集定义存在 $\sqrt{3}-1$ 的冗余,即使通过松弛距离值定义活动集,其鲁棒性依旧受到计算环境的影响,为此,文献[6]把活动集定义为式(20),提高了活动集的拓扑分离的可靠性,但同样不能保持活动集的有效性.

$$L_0 = \left\{ x \mid -\frac{\sqrt{3}}{2} \leq u(x) \leq \frac{\sqrt{3}}{2} \right\} \quad (17)$$

在目前的稀疏场算法实现中,外层体素集实际上是根据拓扑关系建立的,即定义和实现不一致.另外,稀疏场算法简单地通过加减一个体素边长求边界曲面周边外层体素集内的体素 u 值,导致相当的误差,在形状过渡的中后期使边界等值面变得粗糙,需要合适的反走样算法进行平滑处理.

本文针对以上情况做了如下改进:

根据距离值符号关系(即边界内外的拓扑关系)给出了单侧活动集的定义;

外层体素集统一用拓扑关系进行定义;

通过从内到外的距离计算,提高了进化算法效率;

提出了均值平移和窄带回退两种反走样算法,前者比较简单、快速;后者失真度较低.

2 改进的稀疏场算法

2.1 单侧活动集定义

对于三维空间,式(17)定义的体素集能够实现了对边界曲面的超覆盖,并对 Ω 进行拓扑 6 分,模型内部和外部空间相互不存在 6 连通.在这个定义中,活动集的最大厚度为 $\sqrt{3}$.最有效且确保拓扑 6 分的活动集厚度应该是 1,其定义根据 u 值的符号来确定.

记一个体模型为 $V=\{X,u\}$, X 是所有体素中心点的集合, u 是 X 中每个元素的距离值;单位向量 $e_1=\{1,0,0\}$, $e_2=\{0,1,0\}$, $e_3=\{0,0,1\}$,每个体素的 6 连通邻居定义为

$$C(x_{i,j,k}) = \{x_{l,m,n} \mid (l,m,n) = (i,j,k) \pm e_p, 1 \leq p \leq 3\} \quad (18)$$

定义体素(距离值)关系算子为

$$\begin{aligned} A(x_{i,j,k}, x_{l,m,n}) &= \begin{cases} \text{TRUE}, & \text{for } x_{l,m,n} \in C(x_{i,j,k}) \\ \text{FALSE}, & \text{for } x_{l,m,n} \notin C(x_{i,j,k}) \end{cases} \\ O(x_{i,j,k}, x_{l,m,n}) &= \begin{cases} \text{TRUE}, & \text{for } u(x_{i,j,k}) \times u(x_{l,m,n}) \leq 0 \\ \text{FALSE}, & \text{for } u(x_{i,j,k}) \times u(x_{l,m,n}) > 0 \end{cases} \end{aligned} \quad (19)$$

文献[6]中提出如式(20)所示的基于拓扑关系的活动集定义,但受限于活动集必须覆盖零水平集的束缚,其活动集厚度为 2.

$$ActiveSet' = \{x_{i,j,k} \mid \exists x_{l,m,n} (A(x_{i,j,k}, x_{l,m,n}) \wedge O(x_{i,j,k}, x_{l,m,n}))\} \quad (20)$$

本文打破该束缚,提出如式(21)所示的活动集定义:

$$ActiveSet = \{x_{i,j,k} \mid u(x_{i,j,k}) \geq 0, \exists x_{l,m,n} (A(x_{i,j,k}, x_{l,m,n}) \wedge O(x_{i,j,k}, x_{l,m,n}))\} \quad (21)$$

式(21)的定义,把活动集定于零水平集的单侧,活动集厚度为 1,实现了活动集厚度的极小化,可以显著提高进化速度.

2.2 外层体素集定义

为描述方便,把 $ActiveSet$ 记为 L_0 ,而外层体素记为 $L_{\pm p}$,其中 p 是自然数,正值表示在活动集内部;负值表示在活动集外部.根据拓扑连通关系, L_{-1} 和 L_1 定义为

$$L_{-1} = \{x_{l,m,n} \mid x_{i,j,k} \in L_0, A(x_{i,j,k}, x_{l,m,n}), u(x_{i,j,k}) < 0\} \quad (22)$$

$$L_1 = \{x_{l,m,n} \mid x_{i,j,k} \in L_0, A(x_{i,j,k}, x_{l,m,n}), u(x_{i,j,k}) \geq 0\} \quad (23)$$

根据拓扑连通关系,其他各外层体素集定义为

$$L_p = \{x_{l,m,n} \mid x_{i,j,k} \in L_{p-1}, x_{l,m,n} \notin L_{p-2}, A(x_{i,j,k}, x_{l,m,n})\} \quad (24)$$

$$L_{-p} = \{x_{l,m,n} \mid x_{i,j,k} \in L_{-p+1}, x_{l,m,n} \notin L_{-p+2}, A(x_{i,j,k}, x_{l,m,n})\} \quad (25)$$

这样,得到了满足水平集进化需要的拓扑严格的外层体素集.

2.3 算法流程

为每个体素设置两个属性:一个是当前层编号 *layer*,如-1,0,1 分别表示其属于 L_{-1}, L_0, L_1 ;另一个是过程中层编号 *status*,表示这次进化结束后其将所属的层,编号值为 *OUT* 表示该体素不在稀疏场体素集内.*layer* 的值由算法中类似于步骤(c)的函数初始化,所有体素 *status* 的初始值全部为 *OUT*.设单向外层体素共 N 层,算法流程如下:

(a) 依据式(11)更新 L_0 中每个体素的 u 值.

(b) 依据式(26)、式(27)分别更新 $L_{\pm 1}$ 中体素的 u 值.

$$u(x_{i,j,k}) = \min(u(x_{l,m,n}) + 1), x_{i,j,k} \in L_p, x_{l,m,n} \in C(x_{i,j,k}) \cap L_{p-1}, \quad \text{for } p > 0 \quad (26)$$

$$u(x_{i,j,k}) = \max(u(x_{l,m,n}) - 1), x_{i,j,k} \in L_p, x_{l,m,n} \in C(x_{i,j,k}) \cap L_{p+1}, \quad \text{for } p < 0 \quad (27)$$

(c) 判断 L_{-1}, L_0 和 L_1 中的每个体素 x 是否满足依据式(21).如果满足,则 *status*(x)赋值为 0.

(d) 以 L_0 体素为基础,依据式(22)、式(23)重建 L_{-1}, L_1 ,体素 u 值依据式(26)、式(27)计算.

(e) 以内层体素为基础,由内向外,依据式(24)、式(25)重建其他外层体素集,体素 u 值依据式(26)、式(27)计算.

(f) 对于模型中的每个体素 x ,如果 *status*(x)不是 *OUT*,则把 *layer*(x)赋值成 *status*(x),并把 *status*(x)设置为 *OUT*;否则,如果 *layer*(x)不等于 *OUT*,则设置 *layer*(x)为 *OUT*.

(g) 当步骤(a)中计算得到的进化因子 γ_D 的最大值小于目标模型体素宽度时,结束进化.

2.4 性能改进

由于在稀疏场算法中需要大量访问体素的 6 连通邻居,其邻居属性的快速定位是提高时间性能的关键.以前算法中,运用链表存储各体素集属性,搜索时间很长,所以本文采用与体素数量对应的数组.由于 *layer* 和 *status* 的数值范围很小,用 4 位表示已经足够(根据前面水平集进化算法,2 位即可满足要求,4 位可以保留冗余方便平滑算法实现),所以本文用一个字节保存 *layer* 和 *status* 值,所需内存空间较小.

在稀疏场算法计算过程中,其时间性能大致是 $O(w^3)$,其中 w 是稀疏场厚度.在这个过程中,有 3 个环节对计算的时间性能影响比较大:

活动集体素的数量,因为每个活动体素需要计算其 Δu ;

在计算过程中需要访问 6 连通邻居的操作数量,每个这样的操作,需要计算邻居的属性地址并定位;属性的读取和赋值,每次读取和赋值都需要在长数组中进行定位.

基于这 3 个因素,本文算法从几个环节改进了时间性能:

在 L_0 定义为式(17)时,其 $w \approx \sqrt{3} + 2N$,其中 N 是单侧外层体素集的层数;当 L_0 定义为式(20)时, $w \approx 2 + 2N$;在 L_0 定义为式(21)时,其 $w \approx 1 + 2N$,活动集体素数量减少;

外层体素集的重建实行由内向外的扩散策略,这样避免了最外层体素 6 连通邻居的访问;

从内向外的计算过程中利用了原有的 $u, layer$ 和 *status* 值,避免了大量的赋值操作.

在 3 层体素稀疏场中,实验结果证实时间性能明显提高.另外,运用拓扑关系定义和计算各体素集,确保了活动集的拓扑 6 分性质.而用距离值范围定义活动集,需要采用相当大的距离值范围,以避免计算误差引起的错误,而且缺乏理论上的鲁棒性.

3 反走样算法

在稀疏场算法中,对活动集以外的体素 u 值的计算采用式(26)和式(27)的方式,这样提高了计算效率.但是,这种近似值存在较大的误差,这种误差引起的走样在形状过渡的中后期尤其明显,如后文图 1 中的(f)和(g),表面相当粗糙.为此,本文提出了两种平滑算法:均值平移和窄带回退.

3.1 均值平移

均值平移是把体素的 u 值与邻居的 u 值进行平均,在平均计算完成后,由于等值面内外相邻体素数量和 u 值不对称,均值计算后产生一定的偏移,需要进行平移,回退到原始边界曲面附近。 L_{-1}, L_0 和 L_1 是用 Marching Cubes 方法从体模型中求取边界网格模型的数据基础,所以需要对这些 3 层进行均值计算。

为了确保 L_{-1}, L_0 和 L_1 数据均值计算后的精度,首先要求外层体素集层数 $N \geq 2$,即至少建立 L_{-2} 和 L_2 。其次需要克服式(26)和式(27)计算外层体素集的误差。所以,本文采用快进法(fast marching)重新对 L_0 以外的体素集 u 值进行计算,下面是算法流程。

(a) 如果稀疏场外层体素集不满足 $N \geq 2$,依据式(24)和式(25)建立最小稀疏场。

(b) 用快进法求外层体素集 u 值,提高精度。

(c) 进行均值计算的关键一步。先对控制变量进行说明: $offset$ 是个体素求均值后 u 值的偏移; $offsetAvg$ 是一次循环中所有体素 $offset$ 的均值; $offsetAvgLast$ 是上一次循环的 $offsetAvg$; $breakLine$ 是循环结束条件, $breakLine$ 的值在第 2 次循环结束时依据式(28)求出:

$$breakLine = |(offsetAvgLast - offsetAvg) \times \varepsilon| \quad (28)$$

其中 ε 是较小的经验数字,一般在 0.1 和 0.01 之间, ε 的大小和循环次数成反比。

下面是每次循环的过程:

1. $offsetAvgLast = offsetAvg, offsetAvg = 0$ 。

2. 对每个 L_{-1}, L_0 和 L_1 中的每个 x ,依据式(29)求其均值,并存放于临时数组 $Update$ 中。

$$u_{new}(x) = (u(x) + \sum_{y \in C(x)} u(y)) / 7, y \in C(x) \quad (29)$$

计算 $offset = u_{new}(x) - u(x)$,在 $offsetAvg$ 上加 $offset$ 。

3. $offsetAvg$ 除以 L_{-1}, L_0 和 L_1 中体素数量。

4. 对 L_{-1}, L_0 和 L_1 中的每个体素 x ,依据式(30)更新 u 值。

$$u(x) = Update(x) - offsetAvg \quad (30)$$

5. 如果 $|OffsetAvg - offsetAvgLast| < BreakLine$,则循环结束。

(d) 对每个 L_{-1}, L_0 和 L_1 中的体素 x ,更新 $u(x) = u(x) + back$ 。其中 $back$ 是一个回退平移经验值,约等于体素宽度的一半。

均值回退算法的第一个优点是不需要目标模型作为参考,能独立完成计算;其次是平滑效果明显。缺点是在模型形状突变边缘失真度高,而且 ε 和 $back$ 的值在一定程度上需要根据经验来确定。所以,本文还提供了窄带回退平滑算法。

3.2 窄带回退

稀疏场算法的误差来自对外层体素 u 值计算的简化。如果能够在稀疏场进化中间模型上进行窄带计算,一方面可以让中间模型平滑,这是过渡所需要的;另一方面会使中间模型向目标模型进化,那样会产生新的形变,有时候需要克服。所以,本文先利用窄带算法对中间模型进行平滑,而后根据活动集体素在进化中所运用的目标模型的进化因子 γ_D 进行回退。

(a) 依据式(31)~式(34)对 L_{-1}, L_0 和 L_1 中的体素进行加权统计。

$$forceMinus = \sum \gamma_D(x), \quad \text{for } \gamma_D(x) < 0 \quad (31)$$

$$forcePlus = \sum \gamma_D(x), \quad \text{for } \gamma_D(x) \geq 0 \quad (32)$$

$$uMinus = \sum \gamma_D(x) \times u(x), \quad \text{for } \gamma_D(x) < 0 \quad (33)$$

$$uPlus = \sum \gamma_D(x) \times u(x), \quad \text{for } \gamma_D(x) \geq 0 \quad (34)$$

(b) 运用快进法建立窄带,进行窄带进化。

(c) 依据式(33)和式(34)再次对 L_{-1}, L_0 和 L_1 中的体素进行加权统计,得到 $uMinusNew$ 和 $uPlusNew$ 。

(d) 依据式(35)和式(36)计算单位进化因子偏移量。

$$offsetMinus = \frac{uMinusNew - uMinus}{|forceMinus|} \tag{35}$$

$$offsetPlus = \frac{uPlusNew - uPlus}{forcePlus} \tag{36}$$

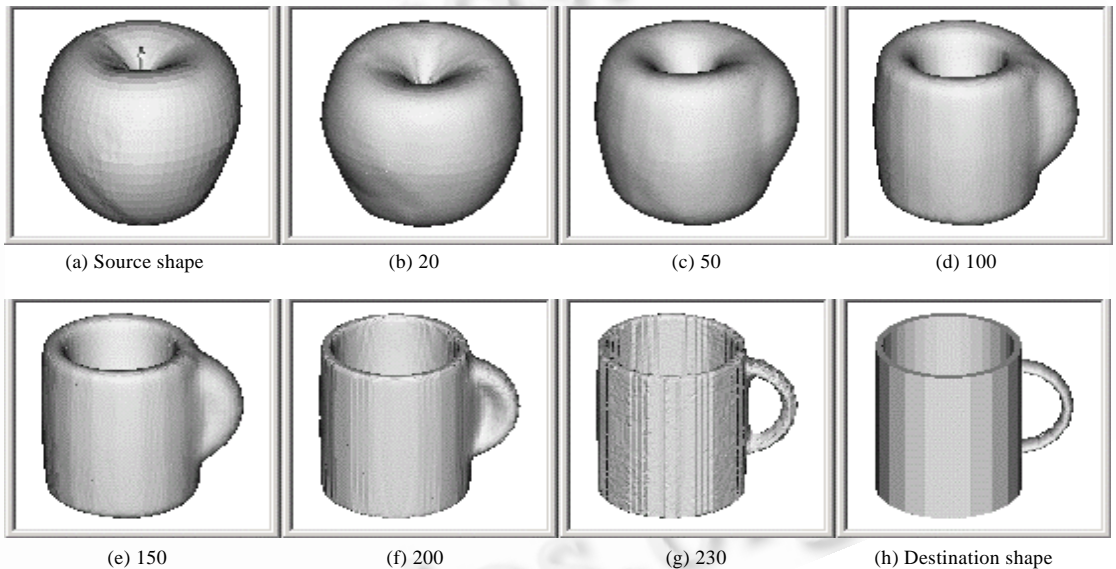
(e) 依据式(37)和式(38)对 L_{-1}, L_0 和 L_1 中的体素进行回退.

$$u(x) = u(x) + offsetMinus \times \gamma_D(x), \text{ for } \gamma_D(x) < 0 \tag{37}$$

$$u(x) = u(x) - offsetPlus \times \gamma_D(x), \text{ for } \gamma_D(x) \geq 0 \tag{38}$$

4 实例分析

实验中比较了距离值范围和拓扑单侧活动集定义下的稀疏场算法,两者属性的存储都采用与体素相对应的数组.图 1 是一个苹果向一个杯子过渡的实例,体模型分辨率为 $128 \times 128 \times 128$,体空间大小分别是模型最小正外包立方体(边平行于坐标轴)的 1.1 倍和 2 倍.运用 Macrching Cube 方法,从稀疏场算法中间体模型求得网格模型用于显示.实验环境:1.4G 的奔IV(M)CPU,352M 内存.



*The number under every picture is the step of the evolution. The evolution was ended at step 249 when the max γ_D became stable which was a little different from the theoretical flow.

Fig.1 The morphing process from apple to cup

图 1 苹果到杯子的形状过渡

表 1 是该实例中的一些指数,验证了稀疏场算法的改进,总体速度明显提高,活动集体素数量和更新时间也相应减少,而内外层体素数量基本相同.

Table 1 Comparison of evolving statistics

表 1 进化指标对照表

Statistic	Distance value	Distance sign (double sides)	Distance sign (single sides)
Evolution steps	251	262	248
L_0 voxel number	34 074	38 046	18 430
L_1 voxel number	18 099	17 351	16 698
L_{-1} Voxel Number	19 862	19 802	19 044
L_0 update time (ms)	80 926	87 726	56 180
L_{-1}, L_1 update time (ms)	45 565	52 815	35 911
Total time ratio	1.374	1.526	1

*Voxel number is the average number of voxels at all steps.

图2和图3分别显示了均值平移和窄带回退的平滑效果.其中:均值平移算法中的 ε 为0.05, $back$ 为体素宽度的一半;窄带回退算法通过快进法建立的窄带单侧外层体素是4层.图2和图3显示了均值平移效果更为平滑.对于杯口这种原本形状就比较锐利的地方,窄带算法保真性好,但是窄带平滑的一进一退也有误差,必要时需要根据经验进行适当的平移.

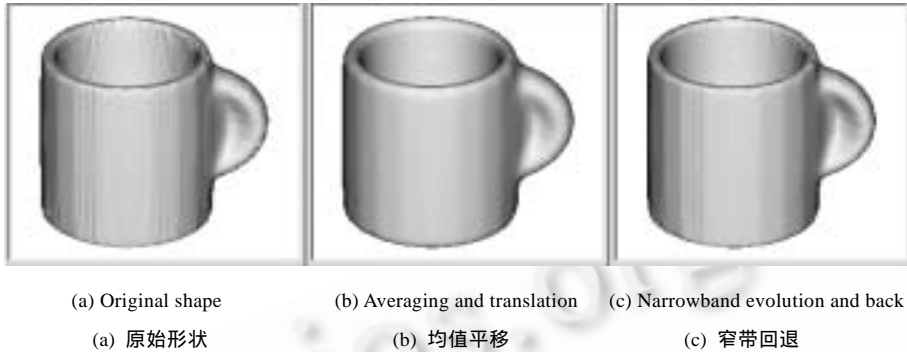


Fig.2 Smoothing of shape Fig.1(f)

图2 图1(f)模型的平滑处理

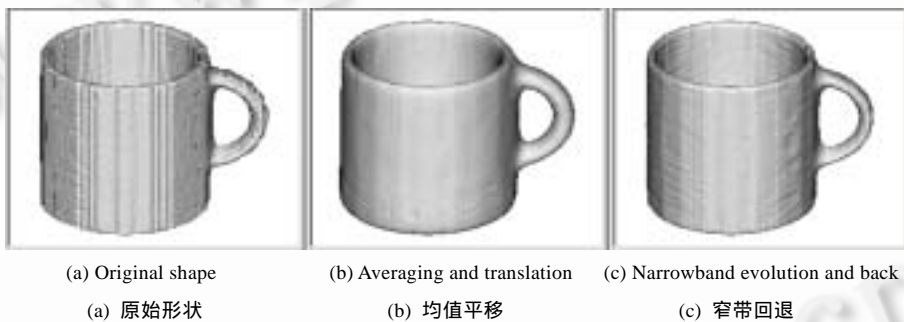


Fig.3 Smoothing of shape Fig.1(g)

图3 图1(g)模型的平滑处理

5 结 论

水平集进化是实现拓扑无关形状过渡的良好方法,稀疏场较之窄带算法更有效,通过反走样算法的运用,可以弥补稀疏场算法的误差失真.限制速度的主要瓶颈是大量6连通邻居操作,如果能够对此有更好的优化,速度还可以有所提高.在平滑算法中,参数的设置根据一定经验来确定,如果能够有严格定义,算法鲁棒性会更强.另外,在进化过程中,如果能够对模型局部实现控制,可以提高该方法在产品中的适用性.水平集算法的历史比较长,其研究已经处于应用领域推广和高效算法完善阶段.文献[12]提出通过p-Laplacian方程控制形状过渡,其思路与水平集算法有类似之处.目前,其数值算法尚缺乏充分研究,可望通过分析水平集算法的高效解决而更深入一步.

References:

- [1] Pan YH. The synthesis reasoning. Pattern Recognition and Artificial Intelligence, 1996,9(3):201-208 (in Chinese with English abstract).
- [2] Kent JR, Carlson WE, Parent RE Shape transformation for polyhedral objects. Computer Graphics, 1992,26(2):47-54.
- [3] Leros A, Garfinkle CD, Levoy M. Feature-Based volume metamorphosis. In: Mair SG, Cook R, eds. SIGGRAPH. Los Angeles: ACM Press, 1995. 449-456.

- [4] Osher SJ, Sethian JA. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 1988,79(1):12-49.
- [5] Adalstein D, Sethian J. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 1994,118(2):269-277.
- [6] Whitaker R, Breen D, Museth K. A level-set approach to 3D reconstruction from range data. *Int'l Journal of Computer Vision*, 1998,29(3):203-231.
- [7] Sethian JA. A fast marching level set method for monotonically advancing fronts. *Proc. of the National Academy of Sciences of USA*, 1996,93(4):1591-1595.
- [8] Breen D, Whitaker R. A level-set approach for the metamorphosis of solid models. *IEEE Trans. on Visualization and Computer Graphics*, 2001,7(2):173-192.
- [9] Museth K, Breen D, Whitaker R, Barr A. Level set surface editing operators. *ACM Trans. on Graphics*, 2002,21(3):330-338.
- [10] Houston B, Wiebe M, Batty C. RLE sparse level sets. In: *SIGGRAPH on Sketches & Applications*. Los Angeles, 2004. 1-1. http://www.exocortex.org/siggraph/Houston2004_RleSparseLevelSets.pdf
- [11] Cohen-Or D, Kaufman A. Fundamentals of surface voxelization. *Graphics and Image Processing*, 1995,57(6):453-461.
- [12] Cong G, Esser M, Parvin B, Bebis G. Shape metamorphosis using p-Laplacian equation. In: Kittler J, Petrou M, Nixon M, eds. *Proc. of the IEEE Int'l Conf. on Pattern Recognition*. Cambridge: IEEE Computer Society Press, 2004,4:15-18.

附中文参考文献:

- [1] 潘云鹤.综合推理的研究.模式识别与人工智能.1996,9(3):201-208.



翁建广(1975 -),男,博士,主要研究领域为多媒体通信,计算机图形学,智能 CAD,地理信息系统.



潘云鹤(1946 -),男,教授,博士生导师,中国工程院院士,CCF 高级会员,主要研究领域为人工智能,计算机图形学,CAD,形象思维,计算机美术.



庄越挺(1965 -),男,博士,教授,博士生导师,主要研究领域为多媒体分析与检索,计算机动画,人工智能.