

基于奔腾 SIMD 和分割技术的快速体绘制

袁非牛 诸葛斌 周荷琴 冯焕清

(中国科学技术大学信息科学与技术学院, 合肥 230027)

摘要 成像速度是影响体绘制应用的关键。为了提高成像速度, 提出一种基于 Intel 奔腾 SIMD 和分割技术的快速体绘制算法。仅仅应用奔腾 SIMD 并行技术, 常规光线投射算法的成像速度能够提高 2~5 倍。奔腾 SIMD 并行指令与分割技术相结合, 减少了大量空采样, 进一步提高了成像速度, 而且这种简单的分割技术能够快速适应转换函数的改变。在一台 P4/1.6G 的 PC 机上, 以 512×512 分辨率渲染时, 该算法渲染速度比常规光线投射算法提高了 10 多倍, 使得等值面的体绘制速度能够达到 1~3 帧/秒。实验结果表明, 该算法具有渲染速度快、成像质量高等显著优点, 而且不需要费时的预处理和特殊体视硬件, 具有较大的实际应用价值。

关键词 计算机图象处理 (520 · 6040) SIMD MMX/SSE/SSE2 可视化 光线投射 体绘制

中图法分类号: R319 **文献标识码:** A **文章编号:** 1006-8961(2003)12-1438-06

A Fast Volume Rendering Algorithm Based on Intel SIMD and Segmentation Technologies

YUAN Fei-niu, ZHU-GE Bing, ZHOU He-qin, FENG Huan-qing

(University of Science & Technology of China, School of Information Science & Technology, Hefei 230027)

Abstract Rendering speed is a key technology of volume rendering. In order to accelerate rendering process, we present a novel imaging acceleration approach based on Intel SIMD and segmentation technologies, which obtain significant speedup without degrading image quality. Only applying Intel SIMD techniques, we can obtain the rendering speed 2~5 times faster than brute force ray casting. SIMD techniques are combined with threshold segmentation to skip large empty samplings, and the rendering speed is further more improved. Because the time of threshold segmenting a large data set is very short, the algorithm can quickly display the rendered image once the threshold values are changed. Experiments have been done on a single processor P4/1.6G PC, and it is 10 more times faster than brute force ray casting and about 1 to 3 frames per second is obtained with the image size 512×512 . Our algorithm has some advantages that rendering speed can be greatly accelerated without any specialized purpose hardware and time-consuming preprocessing, and images quality is very high.

Keywords Computer image processing, Single-Instruction Multiple-Data, MMX/SSE/SSE2, Visualization, Ray casting, Volume rendering

0 引言

体数据可视化技术被广泛地应用在医疗诊断、流场计算、地理勘探等领域, 其可视化方法主要有面绘制和体绘制两大类^[1~3]。面绘制是先根据给定的阈值, 采用某种算法(如移动立方体、小球跳跃法等)提取等值面(Isosurfaces), 然后再用传统多边形绘

制算法进行渲染, 也可利用图形硬件进行渲染, 因而能够获得较快的绘制速度, 但其在提取等值面片的过程中丢失了体数据中的大量细节信息, 绘制的图象质量不高, 而且提取的等值面片数量一般比较大, 有时甚至会超出图形硬件的处理能力。此外, 当阈值改变时, 还必须重新进行费时的等值面片提取过程。而体绘制不必把等值面转化为中间几何图元的表示形式, 而是根据特定的转换函数直接进行绘

基金项目: 安徽省自然科学基金(01042202)

收稿日期: 2003-04-10; 改回日期: 2003-05-27

制,可以表现体数据中的细腻结构,所以体绘制更易
适合对图象质量要求较高的应用场合,但是成像速
度慢。

光线投射算法是一种典型的图象空间体绘制算
法^[2],能够生成高质量图象,但该算法还很难在 PC
机上达到交互的渲染速度.为了加快成像速度,许多
学者提出了许多不同的加速技术.其中一类基于空
间跳跃(Space Leaping)^[4~6]和特殊数据结构(如八
叉树等)的加速技术,能够获得较高的加速比,但需
要费时、复杂的预处理过程;另一类加速技术是在图
象质量和成像速度之间寻求折衷^[7],但图象质量的
降低影响了应用效果;还有一类就是采用专用硬件
(如 Volume Pro^[8]等)、机群^[9]或并行计算机^[10]等技
术,采用这些技术虽然能获得较高的绘制速度,但系
统价格昂贵、代价高。

机群或多处理器等技术可以有效地加速投射光
线间的粗粒度并行问题,但不适合解决投射光线内
的细粒度并行问题.而 SIMD 技术非常适合加速光
线内的细粒度并行计算问题.另外,应用简单的分割
技术能够避免大量的空采样,大大加快了成像速度.
奔腾 SIMD 和分割技术相结合,使得成像速度在总
体上比常规光线投射算法快了 10 多倍!

1 奔腾 SIMD 技术

Intel 奔腾系列微处理器为了加快多媒体、数值
计算、语音识别等应用的执行速度,采用了单指令,
多数据并行加速技术(Single-Instruction,
Multiple-Data, SIMD).SIMD 技术的核心就是一
条指令能同时完成多条相同的操作,典型的 SIMD
操作如图 1 所示,其中操作 Op 可以是普通的加、
减、乘、除算术运算,还可以是各种逻辑运算等^[11].
为了进一步优化执行速度,Intel 奔腾微处理器还把
常用的复杂计算(如开方运算等)采用硬件实现,使

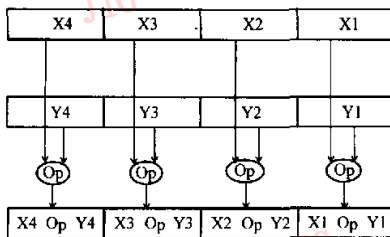


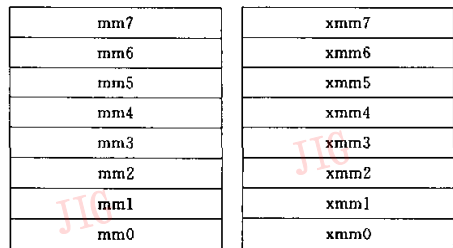
图 1 典型 SIMD 操作

得一条 SIMD 指令可以完成多条复杂的运算,这样
大大地加快了指令的执行速度。

Intel IA-32 构架的 SIMD 技术主要由 MMX、
SSE、SSE2 3 部分组成,其中 MMX 技术是伴随奔
腾微处理器诞生而产生的,随后 P3 处理器引入
SSE(Streaming SIMD Extensions,流 SIMD 扩展)
技术,P4 则引入了 SSE2(Streaming SIMD
Extensions 2,流 SIMD 扩展 2)技术。

1.1 MMX 技术

MMX 技术引入了 8 个 64 位的 MMX 寄存器
(图 2(a)).MMX 寄存器实际上是 x87 协处理器 8
个浮点寄存器的别名.采用 MMX 指令能够并行执
行紧缩 8 个字节、4 个字、2 个双字等类型数据运算
操作.这些指令大大地提高了多媒体和通讯应用软
件的性能,但是它只能处理定点数,不能完成浮点数
操作^[12].



(a) 64-位 MMX 寄存器 (b) 128-位 XMM 寄存器

图 2 SIMD 寄存器

1.2 SSE 技术

SSE 技术引入 8 个全新的 128 位 XMM 寄存器
(图 2(b)),能够并行执行 4 个紧缩单精度浮点数操
作,也能够完成 64 位整型数的操作,同时引入了数
据预取指令、滞后存储指令、以及其他高速缓存和内
存调整指令等.这些技术非常适合应用在 3D 图形
渲染、语音识别、视频编码和解码等场合.但 SSE 还
不能处理双精度浮点数、数据类型转换和紧缩整型
数等操作。

1.3 SSE2 技术

SSE2 技术没有再增加新的寄存器,但增加了紧
缩双精度浮点数操作,同时增加了对紧缩 16 个字
节、8 个字、4 个双字和 2 个 64 位 4 字整型数的操
作.另外,SSE2 增加了各种数据类型的转换指令,扩
展了数据交换操作和数据高速缓存等操作.这些技
术很适合应用在 3D 图形渲染等场合。

2 指令级并行化及分割加速技术

在光线投射成像算法中,光线间和光线内均存在并行性.光线间的并行性粒度大,一般适合采用机群或多处理器等解决方案.光线内的并行性粒度细(如每条光线上的重采样计算、法向量的密度梯度近似等),非常适合采用 SIMD 指令并行加速技术.分割技术可以减少不必要的空采样,进一步提高成像速度.

2.1 体绘制的成像理论及并行性

在体绘制中,体数据被看成是由许多小颗粒组成,这些颗粒既可以吸收光源发出的光,同时又能根据颗粒的高光、漫反射等材质属性反射特定光到观察者眼中.根据这个理论,沿着特定视线方向所观察到的颜色实际上是该视线穿过的所有颗粒的光累积效果.沿视线采用不同的累积方式,就可以得到不同的成像结果.常用的累积算子有 Over 或 Under、Attenuate 以及 MIP 三大类.Over 或 Under 算子产生的图象符合真实世界的物体消隐和颜色融合规律,经常被应用在 Splatting、Shear-warp、光线投射等算法中. Attenuate 算子产生的图象和 X-Ray 成像效果类似. MIP 算子经常被应用在最大密度投影算法中(如血管造影中).本文主要讨论应用 Over 或 Under 算子的光线投射算法加速问题,其累积计算如式(1)所示^[13].

$$I_{\lambda}(x, r) = \int_0^L C_{\lambda}(s)u(s)e^{-\int_0^s \mu(\omega)ds} ds \quad (1)$$

其中, L 是视线 r 的长度, $I_{\lambda}(x, r)$ 是沿着从图象平面上 x 点发出的视线 r 做积分得到波长为 λ 的光的累积强度, $C_{\lambda}(s)$ 是在视线 r 上 s 处的反射光颜色, $u(s)$ 是在视线 r 上 s 处的衰减系数.式(1)是发射与吸收体绘制光学模型,它是一个沿着 3D 直线作连续积分的运算,因此效率低,而且不便于计算机进行离散处理.为此,对式(1)进行离散和近似处理:先对式中的积分作离散化处理,然后将指数部分按 Taylor 级数展开并作近似处理,最后得到色彩合成公式^[13]

$$I_{\lambda}(x, r) = \sum_{i=0}^{L/\Delta s - 1} C_{\lambda}(i\Delta s)\alpha(i\Delta s) \times \prod (1 - \alpha(j\Delta s)) \quad (2)$$

其中, $\alpha(i\Delta s) = u(i\Delta s)\Delta s$ 为不透明度, Δs 为采样间隔. $\alpha(s)$ 和 $C_{\lambda}(s)$ 由转换函数(Transfer function)、插

值方式、光照和着色模型共同决定.

根据以上分析,可以编制出光线投射体绘制算法流程如图 3 所示.

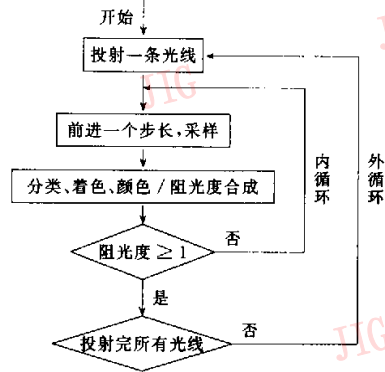


图 3 算法流程

从图象平面上一像素出发,投射一条光线,沿着这条光线以等间隔 Δs 进行重采样^[14],然后分类、着色,最后根据式(2)合成所有采样点的贡献.假设图象大小为 $n \times n$,每条光线重采样数为 n ,则常规光线投射算法的时间复杂度为 $O(n^3)$,因而成像速度非常慢.幸运的是,各条光线上的计算彼此相互独立,可以并行处理.因此这类外层循环(即光线间,图 3 所示)比较适合采用机群等解决方案.同时,同一条光线上的采样计算、颜色合成等也存在并行性,但这类内层循环计算的并行粒度细,显然不适合机群等方式的加速,但是非常适合 SIMD 指令级并行化.由分析可知,内层循环计算量主要集中在重采样和着色阶段,因此所要解决的主要是这两个阶段计算的并行化.

2.2 重采样的并行化

光线每前进一个步长,需要对体数据进行一次重采样(一般采用三线性插值),如图 4 所示.

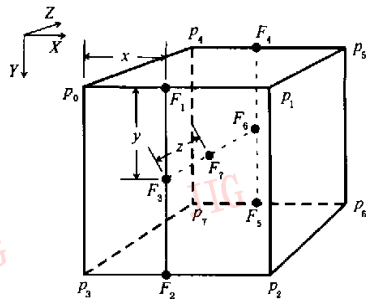


图 4 三线性插值

假定该体素是一单位立方体(边长为 1),已知 8 个角点的体素值(p_0, \dots, p_7),则根据三线性插值方式

$$\begin{aligned} F_1 &= xp_1 + (1-x)p_0 \\ F_2 &= xp_2 + (1-x)p_3 \\ F_3 &= yF_2 + (1-y)F_1 \\ F_4 &= xp_5 + (1-x)p_4 \\ F_5 &= xp_6 + (1-x)p_7 \\ F_6 &= yF_5 + (1-y)F_4 \\ F_7 &= zF_6 + (1-z)F_3 \end{aligned} \quad (3)$$

可以求出该体素内任意点(x, y, z)的重采样值,其中 $F_1 \sim F_7$ 为中间变量, $p(x, y, z) = F_7$.

按照式(3)的计算方式,每次重采样插值必须至少进行 14 次乘法、7 次加法和 7 次减法运算,插值一次消耗了很多的时钟周期数.为此采用 SIMD 指令并行化插值计算.为了适应 SIMD 并行指令格式,把插值计算方式按图 5 进行了调整.

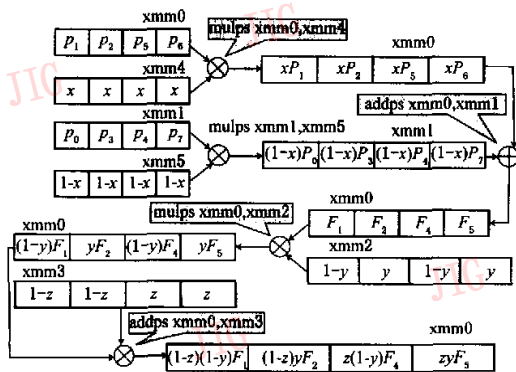


图 5 三线性插值的 SIMD 指令并行化示意图

由于 SSE 技术能够并行处理 4 条单精度浮点数操作,因此把 8 个角点的体素值紧缩成两个 128 位的 XMM 寄存器数据,每个寄存器包含 4 个单精度浮点数.如图 5 所示,寄存器 xmm0 由 p_1, p_2, p_5, p_6 组成,寄存器 xmm1 由 p_0, p_3, p_4, p_7 组成.同样,把坐标系数 x, y, z 紧缩成 xmm2(包含 $1-y, y, 1-y, y$),xmm3(包含 $1-z, 1-z, z, z$),xmm4(包含 x, x, x, x),xmm5(包含 $1-x, 1-x, 1-x, 1-x$).按照图 5 的执行路径计算,并将计算结果存放在寄存器 xmm0 中,最后还需要把 xmm0 的 4 个单精度浮点数沿水平方向相加得到最终结果 $F_7 = (1-z)(1-y)F_1 + (1-z)yF_2 + z(1-y)F_4 + zyF_5$,经过简单验证可知,计算结果和常规插值方式(式(3))完全相同.以下给出三线性插值 SIMD 并行实现的核心汇

编代码:

```
//xmm0(p1, p2, p5, p6), xmm1(p0, p3, p4, p7)
//xmm2(1-y, y, 1-y, y), xmm3(1-z, 1-z, z, z)
//xmm4(x, x, x, x), xmm5(1-x, 1-x, 1-x, 1-x)
mulps xmm0, xmm4; //xmm0 = xp1, xp2, xp5, xp6
mulps xmm1, xmm5; //xmm1 = (1-x)p0, (1-x)p3,
(1-x)p4, (1-x)p7
addps xmm0, xmm1; //xmm0 = F1, F2, F4, F5
mulps xmm0, xmm2; //xmm0 = (1-y)F1, yF2,
(1-y)F4, yF5
mulps xmm0, xmm3; //xmm0 = (1-z)(1-y)F1,
(1-z)yF2, z(1-y)F4, zyF5
//最后水平累加寄存器 xmm0, 得到最终计算结果.
```

可以看出真正计算的代码中只有 4 次乘法和 1 次加法就可以完成三线性插值,大大缩短了计算时间.当然数据结构的调整和计算结果的水平累加都消耗掉了一定的时钟周期数.

2.3 着色与色彩合成的并行化

着色与色彩合成过程的计算量主要集中在法向量插值和颜色/不透明度的合成上.由于法向量包括 3 个坐标分量,因此必须进行 3 次插值才能得到体素内任意点的法向量.对于法向量的插值,可以采用上述类似的 SIMD 并行化方法.在光线投射算法中,颜色与不透明度合成通常采用从前向后的累积方式(图 6).

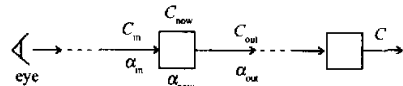


图 6 光线投射颜色与不透明度的合成

因此把式(2)重新改写成如下的递推累积公式

$$\begin{cases} C_{out} = C_{in} + C_{now}\alpha_{now}(1 - \alpha_{in}) \\ \alpha_{out} = \alpha_{in} + \alpha_{now}(1 - \alpha_{in}) \end{cases} \quad (4)$$

在颜色的累积过程中,要防止色彩分量值超过 255,因此采用传统的色彩合成方法每累积一次必须判断累加值是否溢出.频繁的条件测试影响了奔腾硬件的预测命中率和性能.而 MMX, SSE2 的紧缩无符号字节数饱和加法等指令非常适合应用在颜色合成计算上.这种指令对数据的处理比较特殊,如果两个无符号字节数相加超过 255,则结果保持为 255,而不是环绕成小于 255 的数(丢弃进位).

2.4 其他指令级优化技术

光线投射成像算法中,每次插值运算,必须要装载 8 个角点的体素值.频繁地访问内存,将造成整体

性能的巨大损失. 为了使得计算和内存访问时间重叠, 采用 SIMD 软件数据预取指令. 另外, SIMD 拥有 8 个 64 位 MMX 寄存器、8 个 128 位 XMM 寄存器, 可以同时保存多个 32 位的单精度浮点数变量, 因此对于一些常用的变量采用寄存器保存, 这样可以大大地减少内存访问量, 加快了执行速度. 在指令的选取方面, 尽量避免时钟周期数多的指令. 最后对指令的执行顺序进行细致调整, 尽量避免指令的阻塞^[15].

2.5 与分割技术结合

由于在光线投射算法中, 大部分采样是空采样, 即这些对结果图象没有任何贡献作用. 为了避免这类空采样, 对体数据采用简单的阈值分割 (Threshold Segmentation), 即根据转换函数确定上限和下限阈值, 任一像素的 8 个角点的 CT 值均小于下限阈值则属于空腔体素 (Lumen voxels), 均

大于上限则属于背景体素 (Background voxels), 否则属于边界体素 (Boundary voxels). 根据这些分割信息, 只对边界体素内的点进行重采样, 这样就能够避免大量的空采样, 明显提高了成像速度. 由于这种分割方法非常简单, 而且速度快 (如 $512 \times 512 \times 199$ 大小的体数据, 也只需要几秒就可以完成), 因此能够适应随着转换函数的改变而快速调整的需要.

3 实验及结果

在 Visual C++ .NET 的开发环境和一台普通 P4/1.6GHz/256M 的单处理器 PC 机上, 实现了 SIMD 技术加速的光线投射体绘制算法. 对 3 组不同的人体 CT 数据和一组 MR 数据进行实验, 渲染结果图象如图 7 所示.

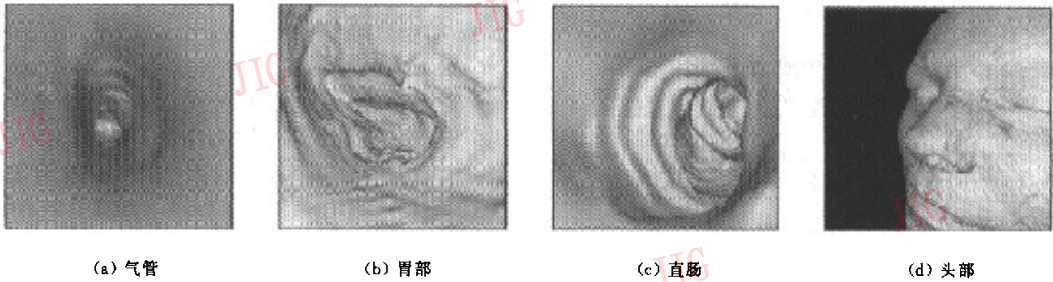


图7 渲染结果图象

实验分别采用人体气管、胃部、直肠和头部数据, 数据大小分别为: $512 \times 512 \times 112$, $512 \times 512 \times 86$, $512 \times 512 \times 199$, $256 \times 256 \times 109$, 其中头部图象是 MR 数据, 对它进行的是“外窥”成像, 而其他的数据都是 CT 数据, 进行的是“内窥”成像.

图 7 对应的 CT、MRI 数据渲染时间如表 1 所示, 可以看出, 采用 SIMD 并行指令优化的算法比常规光线投射快 2~5 倍. 为了减少空采样次数, 进一步提高成像速度, 采用了简单的阈值分割技术, 把整个 3D 体数据分割成空腔体素、边界体素、背景体素三大类. SIMD 技术结合这些分割信息进一步地提

高了成像速度, 使得成像速度总体上提高 10 多倍. 3D 体数据的阈值分割即便是对于较大的体数据 (如 $512 \times 512 \times 199$ 大小), 也只需要几秒就可以完成, 能够快速适应转换函数的改变. 另外, 只要转换函数不变, 就没有必要重新计算分割属性.

4 结论

采用 Intel SIMD 并行指令优化技术, 常规光线投射的成像速度可以提高 2~5 倍. 阈值分割加速技术能够避免大量的空采样, 大大提高了成像速度. SIMD 与分割技术相结合, 使得常规光线投射的成像速度在总体上能提高 10 多倍. 实验结果表明, 在 512×512 分辨率下, 该算法的等值面体绘制速度可以达到 1~3 帧/s. 这种加速技术不需要额外、费时的预处理过程和特殊体视硬件, 只需简单、快速的阈值分割, 就能提高成像速度 10 多倍, 而且成像质量较高, 基本能够满足体数据快速渲染的需要, 具有重

表 1 在图象分辨率 512×512 下的渲染时间

单位: s

数据类型及 渲染部位	常规光线投射 渲染时间	SIMD 加速的光线 投射渲染时间	SIMD 和边界体素 加速渲染时间
CT、气管	4.297	0.844	0.331
CT、胃部	9.250	2.672	0.659
CT、直肠	8.156	1.813	0.450
MR、头部	7.922	3.094	0.696

要的实际应用价值。

参 考 文 献

- 1 Lorensen W, Cline H. Marching cubes: a high resolution 3D surface construction algorithm [J]. *Computer graphics*, 1987, 21(4): 163~169.
- 2 YUAN Feiniu, ZHOU Heqin, ZHAO He *et al.* Sampled points decomposing based ray casting for virtual endoscopy[A]. In: *Proceedings of The 2nd International Conference on Image and Graphic*[C], Hefei, 2002, 4675:1017~1021.
- 3 Lee Westover. Footprint evaluation for volume rendering[J]. *Computer Graphics*, 1990, 24(4): 367~376.
- 4 Anna Vilanova, Eduard Gröller, Andreas Kong. Cylindrical approximation of tubular organs for virtual endoscopy[A]. In: *Proceedings of Computer Graphics and Imaging*[C], Las Vegas, NV, USA; IASTED/ACTA Press, 2000, 11: 283~289.
- 5 Sharghi M, Ricketts I W. A novel method for accelerating the visualization process used in virtual colonoscopy [A]. In: *Information Visualization 2001* [C], San Diego, California, 2001, 167~172.
- 6 Wan Ming, Kaufman Arie, Bryson Steve. High performance presence-accelerated ray casting[A]. In: *IEEE Visualization'99* [C], San Francisco, California, 1999; 379~546.
- 7 Martin Brady *et al.* Two-phase perspective ray casting for interactive volume navigation[A]. In: *IEEE Visualization '97 Conference*[C], Phoenix, Arizona, USA, 1997; 183~189.
- 8 Hanspeter Pfister. Architectures for real-time volume rendering [J]. *Future Generation Computer Systems*, 1999, 15(1): 1~9.
- 9 潘志庚, 钱喜武, 石教英. 并行分布式体绘制算法的设计[J]. *小型微型计算机系统*, 1994, 15(10): 35~39.
- 10 宛铭, 唐泽胜. 改进的 Dividing Cubes 算法及其并行实现[J]. *计算机学报*, 1998, 21(3): 252~260.
- 11 Intel Pentium 4 and Intel Xeon Processor Optimization reference manual, Intel Pentium 4 Processor Overview[EB/OL]. <http://developer.intel.com/design/pentium4/manuals/24896607.pdf>, 2002; 25~52.
- 12 IA-32 Intel® Architecture Software Developer's Manual Volume 2, Instruction Set Reference [EB/OL]. <http://developer.intel.com/design/pentium4/manuals/24547108.pdf>, 2002; 43~826.
- 13 Klaus Mueller, Torsten Moller, Roger Crawfis. Splatting without the blur [A]. In: *IEEE Visualization '99* [C], San Francisco, USA, 1999; 363~371.
- 14 陈礼民, 梁晓辉, 邵志东. 基于数据场相关性的光线投射算法[J]. *中国图象图形学报*, 1999, 4(10): 762~765.
- 15 Intel Pentium 4 and Intel Xeon Processor Optimization reference manual [EB/OL]. *Optimizing for SIMD Floating ~ point Applications*. <http://developer.intel.com/design/pentium4/manuals/24896607.pdf>, 2002; 197~216.



袁非牛 1976年生, 中国科学技术大学博士研究生. 主要研究方向为医学影像图象处理、三维可视化。



请葛斌 1976年生, 2003年获中国科学技术大学生物医学工程博士学位. 主要研究方向为医学影像图象处理、智能交通系统。



周荷琴 1946年生, 中国科学技术大学自动化系教授, 博士生导师. 主要研究方向为医学影像图象处理、计算机网络、智能信息处理等。



冯焕清 1945年生, 中国科学技术大学电子科技系教授, 博士生导师. 主要研究方向为医学影像图象处理、计算机仿真、生物医学工程、智能医疗仪器等。