

快速分形图象编码*

何爱军 马争鸣

(中山大学电子系信息处理实验室, 广州 510275)

摘要 提出了一种新的快速分形图象编码方法。该方法通过自适应的图象分块、相似块集的矩分类、相似块和图象块的正交分解以及象素递增的动态搜索等4个步骤来加快分形图象编码的过程。实验结果表明,提出的方法在缩短编码时间方面确有其效,特别是对于细节丰富的图象,效果尤为明显。

关键词 分形图象编码 自适应图象分割 矩分类 正交投影

0 引言

分形图象编码是一种新的图象编码方法^[1-3],因其具有较高压缩比和解码简单快速而受到图象编码界的关注。但是,分形图象编码的编码时间过长,严重影响了它的实用性。因此,目前分形图象编码的研究大都集中于如何减少编码时间。本文提出一种快速分形图象编码的新方法。这种方法通过自适应的图象块分割、相似块集的矩分类、相似块和图象块的正交分解以及象素递增的动态搜索等4个步骤来加快分形图象编码的过程。

1 快速分形图象编码

1.1 自适应图象分块

分形图象编码是一种分块编码,整幅图象的分形码由各图象块的分形码所组成。图象分块对压缩比、重构图象的质量和编码时间都有直接影响。图象块的尺寸一般为 8×8 或 4×4 。采用固定尺寸的图象分块,虽然比较简单,但它不能充分利用图象本身的特性,因而缺乏灵活性。事实上,随着图象本身的大小和自相似程度的不同,在更大尺寸的图象块上进行分形编码也是完全可能的。因此,有效的方法应是能根据图象本身的特性来进行自适应的、可变尺寸大小的分块,对于图象变化比较缓慢的部分用大块分割,以提高压缩比;对图象细节比较丰富的部分用小块分割,以保证图象恢复质量,从而使整个过程

具有最佳效果。

在图象自适应分块的过程中,先把图象分割成若干个尺寸较大的图象块(如 512×512 的图象可以分割成64个 64×64 的图象块),然后,对这些尺寸较大的图象块进行分形编码。如果某个图象块分形编码的效果不理想(也即与相似块分形匹配时误差太大),就把这个图象块再分割成4个尺寸较小的图象块(如 64×64 的图象块分成4个 32×32 图象块)。如此类推,直至每个图象块都有较为理想的分形编码为止。

图象自适应分块增加了算法的复杂度。但是,但在某个尺寸较大的图象块上分形编码取得成功,则对整幅图象压缩比的提高和编码时间的缩短十分有效。

1.2 矩分类

分形图象编码之所以费时间,是因为需要在一个庞大的相似块集中进行搜索。为了提高搜索的效率,一个常用的方法是把相似块集分类,从而把全局搜索简化为局部搜索。我们利用相似块的一阶矩和二阶矩对相似块集依次进行粗分类和细分类,最终把相似块集分成72类,使得搜索只在其中一个类中进行。

(1) 粗分类

我们把相似块等分为4个子块,计算每个子块的一阶矩: A_1, A_2, A_3, A_4 ,如图1所示。

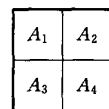


图1

* 本文研究受广东省自然科学基金(No. 960033)的资助

收稿日期:1998-01-19;收到修改稿日期:1998-12-08

其中 $A_i = \frac{1}{N} \sum_{j=1}^N p_{ij}$, N 表示子块象素的数目, p_{ij} 表示子块的象素值, $i=1, 2, 3, 4$. 这4个一阶矩按从大到小顺序排列, 总共有24种不同的排列. 按照8种几何变换(恒等变换, 90° 旋转变换, 180° 旋转变换, 270° 旋转变换, 水平变换, 垂直变换, 左斜对角线变换, 右斜对角线变换), 可以把这24种排列分为3大类, 它们是:

第1类:

$$A_1 \geq A_2 \geq A_3 \geq A_4, A_4 \geq A_1 \geq A_2 \geq A_3, \\ A_3 \geq A_4 \geq A_1 \geq A_2, A_2 \geq A_3 \geq A_4 \geq A_1 \\ A_2 \geq A_1 \geq A_4 \geq A_3, A_4 \geq A_3 \geq A_2 \geq A_1, \\ A_1 \geq A_4 \geq A_3 \geq A_2, A_3 \geq A_2 \geq A_1 \geq A_4$$

第2类:

$$A_1 \geq A_2 \geq A_4 \geq A_3, A_3 \geq A_1 \geq A_2 \geq A_4, \\ A_4 \geq A_3 \geq A_1 \geq A_2, A_2 \geq A_4 \geq A_3 \geq A_1 \\ A_3 \geq A_4 \geq A_2 \geq A_1, A_2 \geq A_1 \geq A_3 \geq A_4, \\ A_1 \geq A_3 \geq A_4 \geq A_2, A_4 \geq A_2 \geq A_1 \geq A_3$$

第3类:

$$A_3 \geq A_1 \geq A_4 \geq A_2, A_2 \geq A_3 \geq A_1 \geq A_4, \\ A_4 \geq A_2 \geq A_3 \geq A_1, A_1 \geq A_4 \geq A_2 \geq A_3 \\ A_1 \geq A_3 \geq A_2 \geq A_4, A_2 \geq A_4 \geq A_1 \geq A_3, \\ A_3 \geq A_2 \geq A_4 \geq A_1, A_4 \geq A_1 \geq A_3 \geq A_2$$

(2)粗分类

为了对上述3个大类进一步细分, 我们计算每个子块的二阶矩: B_1, B_2, B_3, B_4 , 如图2所示.

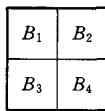


图2

其中 $B_i = \frac{1}{N} \sum_{j=1}^N (p_{ij} - A_i)$, $i=1, 2, 3, 4$. 同样地, 这4个二阶矩按从大到小顺序排列, 总共有24种不同的排列, 每种排列作为一个小类, 故总共有 $3 \times 24 = 72$ 个小类.

当对一个图象块进行编码时, 先判断图象块所属的类别, 然后只在图象块所属的类别中搜索与图象块最佳分形匹配的相似块^①.

1.3 正交分解

在分形图象编码中, 图象块不是直接与相似块进行比较, 而是与相似块仿射变换后的结果进行比

较. 因此, 分形编码实际上是在相似块集与仿射变换集的乘积集中进行搜索. 为了缩小编码时间, 我们不但要考虑相似块集的搜索效率, 也要考虑仿射变换集的搜索效率. 下面, 我们将通过正交分解, 把对仿射变换集的搜索转化为计算图象块与相似块的拟距离, 从而大大缩短了编码的时间.

设 R 和 D 分别表示图象块和相似块. 相似块 D 的仿射变换为: $f(D) = \alpha D + G$, 其中 α 为实数, 称为收缩因子(scalar), G 称为偏移(offset)块. 我们要确定 α 和 G , 使得

$$\|f(D) - R\| = \|(\alpha D + G) - R\| = \min$$

为此, 我们先确定偏移块 G 的选取范围:

$$\Pi = \{G | G = \alpha G_1 + \beta G_2 + \gamma G_3, \\ \alpha, \beta, \gamma \text{ are real numbers} \} \\ = \text{span} \{G_1, G_2, G_3\}$$

其中

$$G_1 = (g_{ij}^{(1)})_{n \times n}, g_{ij}^{(1)} \equiv 1; \\ G_2 = (g_{ij}^{(2)})_{n \times n}, g_{ij}^{(2)} = i; \\ G_3 = (g_{ij}^{(3)})_{n \times n}, g_{ij}^{(3)} = j$$

我们称 Π 为偏移空间. 为了正交分解的方便, 我们利用 Schmidt-Gram 方法, 把 Π 的基 $\{G_1, G_2, G_3\}$ 标准正交化:

$$E_1 = \frac{G_1}{\|G_1\|}, E_2 = \frac{G_2 - \langle G_2, E_1 \rangle E_1}{\|G_2 - \langle G_2, E_1 \rangle E_1\|}, \\ E_3 = \frac{G_3 - \langle G_3, E_2 \rangle E_2 - \langle G_3, E_1 \rangle E_1}{\|G_3 - \langle G_3, E_2 \rangle E_2 - \langle G_3, E_1 \rangle E_1\|}$$

于是, 根据正交投影定理, 我们可以对图象块 R 和相似块 D 进行正交分解:

$$D = D^* + G_D$$

$$\text{其中, } G_D = \sum_{i=1}^3 \langle D, E_i \rangle E_i \in \Pi, G_D \perp D^*;$$

$$R = R^* + G_R$$

$$\text{其中, } G_R = \sum_{i=1}^3 \langle R, E_i \rangle E_i \in \Pi, G_R \perp R^*.$$

显然, $(\alpha D^* - R^*) \perp (G + \alpha G_D - G_R)$. 于是, 根据勾股定理, 我们得到:

$$\|f(D) - R\|^2 = \|(\alpha D + G) - R\|^2 \\ = \|(\alpha D^* - R^*) + (G + \alpha G_D - G_R)\|^2 \\ = \|\alpha D^* - R^*\|^2 + \|G + \alpha G_D - G_R\|^2$$

先选择 α , 使得 $\|\alpha D^* - R^*\|$ 最小:

① 本文所说的相似块集中的相似块都是指经过亚抽样和几何变换后的相似块, 它的尺寸与图象块是一样的.

$$\begin{aligned} \| \alpha D^* - R^* \|^2 &= \langle \alpha D^* - R^*, \alpha D^* - R^* \rangle \\ &= \| D^* \|^2 \alpha^2 - 2 \langle D^*, R^* \rangle \alpha + \\ &\| R^* \|^2 \Rightarrow \alpha_{opt} = \frac{\langle D^*, R^* \rangle}{\| D^* \|^2} \end{aligned}$$

α_{opt} 确定后,再选择 G ,使得 $\| G + \alpha_{opt} G_D - G_R \|^2$ 最小:

$$G_{opt} = G_R - \alpha_{opt} G_D = \sum_{i=1}^3 \langle R, E_i \rangle E_i -$$

$$\alpha_{opt} \sum_{i=1}^3 \langle D, E_i \rangle E_i = \sum_{i=1}^3 \langle R - \alpha_{opt} D, E_i \rangle E_i$$

α_{opt} 和 G_{opt} 确定后,最佳仿射变换 $f_{opt}(D) = \alpha_{opt} D + G_{opt}$ 也就随之确定。

我们称 R 与 $f_{opt}(D)$ 的距离为 R 与 D 的拟距离:

$$\begin{aligned} \rho(R, D) &= \| f_{opt}(D) - R \|^2 \\ &= \| (\alpha_{opt} D + G_{opt}) - R \|^2 \\ &= \| \alpha_{opt} D^* - R^* \|^2 \\ &= \sqrt{\| R^* \|^2 - \frac{|\langle D^*, R^* \rangle|^2}{\| D^* \|^2}} \end{aligned}$$

于是,对一个图象块 R 进行分形编码,就是计算这个图象块与相似块的拟距离,使得拟距离最小的相似块 D ,就是与图象块 R 最佳分形(仿射)匹配的相似块。这个相似块 D 在原图中的位置,经历的几何变换以及 $\alpha_{opt} = \frac{\langle D^*, R^* \rangle}{\| D^* \|^2}$ 和 G_{opt} 在标准正交基 $\{E_1, E_2, E_3\}$ 的系数 $\langle R - \alpha_{opt} D, E_i \rangle, i=1, 2, 3$,就构成了图象块 R 的分形码^{②③}。

1.4 象素递增的动态搜索

如前所述,对一个图象块进行分形编码,就是要计算这个图象块与它所属的类别中每一个相似块的拟距离,这相当于对类中的相似块进行全搜索。在对细节丰富的图象进行分形编码时,相似块集中各个类所包含的相似块,数目还是非常可观的。为了进一步缩短编码时间,我们利用拟距离的某些数学性质,把对类中相似块的全搜索简化为象素递增的动态搜索。

对于任意二个向量 $x = (x_1, \dots, x_k, \dots, x_n), y = (y_1, \dots, y_k, \dots, y_n)$,这里 $1 \leq k \leq n$,我们记

$$x^{(k)} = (x_1, \dots, x_k), y^{(k)} = (y_1, \dots, y_k)$$

$$T(x^{(k)}, y^{(k)}) = \sqrt{\| x^{(k)} \|^2 - \frac{|\langle x^{(k)}, y^{(k)} \rangle|^2}{\| y^{(k)} \|^2}}$$

$$= \sqrt{\sum_{i=1}^k x_i^2 - \frac{|\sum_{i=1}^k x_i y_i|^2}{\sum_{i=1}^k y_i^2}}$$

显然,如果我们把图象块 R 和相似块 D 的正交分量 R^* 和 D^* 按一定顺序写成一维向量的形式:

$$R^* \rightarrow x = (x_1, \dots, x_N), D^* \rightarrow y = (y_1, \dots, y_N)$$

N 表示块所包含象素的数目。则 $T(x, y)$ 就是图象块 R 与相似块 D 的拟距离,也即 $T(x, y) = \rho(R, D)$ 。

定理 1^[4] 设 $1 \leq k < N$,则 $T(x^{(k)}, y^{(k)}) \leq T(x^{(k+1)}, y^{(k+1)})$ 。

定理 2^[4] 设 $\epsilon > 0, 1 \leq k < N, x^{(k+1)} = x^{(k)} \cup \{x_{k+1}\} = (x_1, \dots, x_k, x_{k+1}), y^{(k+1)} = y^{(k)} \cup \{y_{k+1}\} = (y_1, \dots, y_k, y_{k+1})$,而且 $\epsilon' = T(x^{(k)}, y^{(k)}) \leq \epsilon$,

(1) 当 $\epsilon < \| x^{(k)} \|^2$ 时,如果 y_{k+1} 满足 $|y_{k+1} - \alpha| \leq \beta$,则 $T(x^{(k+1)}, y^{(k+1)}) \leq \epsilon$;

(2) 当 $\epsilon > \| x^{(k)} \|^2$ 时,如果 y_{k+1} 满足 $|y_{k+1} - \alpha| \geq \beta$,则 $T(x^{(k+1)}, y^{(k+1)}) \leq \epsilon$;

(3) 当 $\epsilon = \| x^{(k)} \|^2$ 时,如果 y_{k+1} 满足 $2x_{k+1}y_{k+1} \langle x^{(k)}, y^{(k)} \rangle \geq x_{k+1}^2 \| y^{(k)} \|^2 - |\langle x^{(k)}, y^{(k)} \rangle|^2$,则 $T(x^{(k+1)}, y^{(k+1)}) \leq \epsilon$ 。

$$\text{这里 } \alpha = x_{k+1} \frac{\langle x^{(k)}, y^{(k)} \rangle}{\| x^{(k)} \|^2 - \epsilon^2},$$

$$\beta = \frac{\| y^{(k)} \|^2 \sqrt{(\| x^{(k+1)} \|^2 - \epsilon^2)(\epsilon^2 - \epsilon'^2)}}{\| x^{(k)} \|^2 - \epsilon^2}$$

根据这两个定理,我们可以把对类中相似块的全搜索简化为象素递增的动态搜索,具体步骤如下:

第 1 步:把图象块 R 和相似块 D 的正交分量 R^* 和 D^* 按一定顺序编号,如按 Zig-Zag 的顺序,写成一维向量的形式:

$$R^* \rightarrow x = (x_1, \dots, x_N), D^* \rightarrow y = (y_1, \dots, y_N)$$

N 表示块所包含象素的数目。

第 2 步:确定误差限 ϵ 。理论上说,我们只是考虑类中与图象块的拟距离小于或等于 ϵ 的那些相似块。

第 3 步:确定 $1 \leq k < N$ 作为象素递增的初始值。

第 4 步:用 Φ 表示类中所有那些相似块,它们满足 $T(x^{(k)}, y^{(k)}) \leq \epsilon$ 。根据定理 1,图象块与 Φ 之外的相似块的拟距离必然大于 ϵ ,无需考虑。

第 5 步:剔除 Φ 中所有那些相似块,它们的 y_{k+1} 不满足下面的条件:

② $\langle \cdot, \cdot \rangle$ 表示内积, $\| \cdot \|$ 表示范数,内积与范数的关系是 $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$ 。

③ 本节所涉及的数学知识可以参阅任何一本《泛函分析》的教材。

如果 $\epsilon < \|x^{(k)}\|$, y_{k+1} 不满足 $|y_{k+1} - \alpha| \leq \beta$;

如果 $\epsilon > \|x^{(k)}\|$, y_{k+1} 不满足 $|y_{k+1} - \alpha| \geq \beta$;

如果 $\epsilon = \|x^{(k)}\|$, y_{k+1} 不满足 $2x_{k+1}y_{k+1} \langle x^{(k)}, y^{(k)} \rangle \geq x_{k+1}^2 \|y^{(k)}\|^2 - |\langle x^{(k)}, y^{(k)} \rangle|^2$.

根据定理 2, 图象块与这些相似块的拟距离必定大于 ϵ , 无需考虑, 故从 Φ 中剔除出去。

第 6 步: 如果 $k \neq N-1$, 令 $k = k+1$, 重复第 5 步。

第 5 步和第 6 步构成一个循环。在这个循环中, Φ 包含的相似块愈来愈少, 而且图象块与这些相似块的拟距离, 当循环结束 ($k = N-1$) 时, 也随之计算

出来: $T(x^{(k+1)}, y^{(k+1)}) = T(x^{(N)}, y^{(N)}) = \rho(R, D)$ 。使得拟距离最小的相似块就是与图象块最佳分形匹配的相似块。

2 实验结果

实验环境: Pentium120、16MRAM 的 PC 机, Win95 平台, VisualC++4.0 编程环境。

实验数据: Lenna (512×512, 8bpp) 图, Boy (512×512, 8bpp) 图。

实验结果如图 3 和图 4 所示。



(a) Lenna(512×512, 8bpp)原图



(b) 用本文方法对 Lenna 图编码后又解码得到的图象
压缩比: 36.52, 信噪比: 28.86, 编码时间: 19s

图 3



(a) Boy(512×512, 8bpp)原图



(b) 用本文方法对 Boy 图编码后又解码得到的图象
压缩比: 11.37, 信噪比: 32.26, 编码时间: 49s

图 4

实验分析:

(1) 与 Lenna 图相比, Boy 图是一幅有许多细节的图象, 为了不丢失细节, 压缩比不能太高, 编码时间也比较长。

(2) 与许多文献提供的结果相比, 本文提供的结果, 编码时间缩短了。例如, 在压缩比和信噪比大致相同的条件下, 文献[3]提供的结果, 对 Lenna ($512 \times 512, 8\text{bpp}$) 图编码, 编码时间为 21s, 比我们的结果 19s 多了 3s, 对 Boy ($512 \times 512, 8\text{bpp}$) 图编码, 编码时间为 75s, 比我们的结果 49s 多了 26s! 因此, 本文提出的快速分形图象编码方法对细节丰富的图象尤为适用。

3 结 论

(1) 分形图象编码是一种分块编码, 整幅图象的分形码由各个图象块的分形码所组成。图象块的尺寸一般为 8×8 或 4×4 ^[2], 但是, 随着图象本身的大小和自相似程度的不同, 在更大尺寸的图象块上进行分形匹配也是完全可能的。本文通过自适应图象分块, 在多尺寸的图象块上进行分形编码, 不但减少了编码时间, 也提高了压缩比。

(2) 分形编码之所以费时间, 是因为它要在一个庞大的相似块集中搜索。为了提高搜索的效率, 一个常用的方法是把相似块集分类, 从而把全局搜索简化为局部搜索。显然, 一个好的分类方法, 应该使各类包含的元素数目大致相当。Jacquin 利用分类矢量编码的方法^[5], 把相似块集分成三类: shade 类、midrange 类和 edge 类。在实践中我们发现, 这种分类方法把绝大多数相似块都分在 edge 类, 因此, 这种分类方法对于提高搜索效率, 效果差强人意。究其原因, 是因为分形编码中的相似块与矢量编码中的码矢性质是不同的, 照搬矢量编码中码书的分类方法, 未必能够产生预期的效果。本文利用相似块的一

阶矩和二阶矩依次进行粗分类和细分类, 把相似块集分成 72 类, 使得每类所包含的相似块数目大致相当, 从而大大提高了搜索效率。

(3) 在分形图象编码中, 图象块不是直接与相似块进行比较, 而是与相似块仿射变换后的结果进行比较。因此, 分形编码实际上是在相似块集与仿射变换集的乘积集中进行搜索。如果相似块集包含 n 个相似块, 仿射变换集包含 m 个仿射变换, 则分形编码要在 $n \times m$ 种可能的组合中搜索最佳的组合, 运算量是很大的。本文利用正交投影的变分性质, 把对最佳仿射变换的搜索简化为对图象块和相似块的正交分解(在仿射变换的偏移空间及其正交补空间上的正交分解), 从而大大减少了搜索的运算量。这与二次函数最小值的搜索, 可以转化为线性方程组的求解是一样的道理。

(4) 为了进一步加快编码过程, 本文利用象素递增的动态搜索方法, 在象素逐个递增的过程中, 不断缩小类中相似块的搜索范围。

(5) 本文提出的快速分形图象编码方法, 虽然稍嫌复杂, 但是, 一份耕耘一份收获。这种方法对于减少编码时间的效果是实实在在的, 特别是对于细节丰富的图象, 效果尤为明显。

参 考 文 献

- 1 Barnsley M F. Fractal Everywhere. New York: Academic Press, 1988.
- 2 Jacquin A E. Fractal Image coding: A review. In: Proceedings of IEEE, 1993, 81(10):1451~1465.
- 3 Fisher Y. Fractal Image Compression: Theory and Application. New York: Springer-Verlage 1994.
- 4 Bani-Eqbal B. Enhancing the speed of fractal image compression. Optical Engineering, 1995, 34(6):1705~1710.
- 5 Ramamurthi B, Gersho A. Classified vector quantization of images. IEEE Transaction on Communication, 1986, 34(11).



何爱军 中山大学通信与信息系统专业硕士研究生。研究方向为多媒体数据处理与传输。



马争鸣 清华大学模式识别与智能控制专业博士, 现为中山大学电子系副教授。主要研究方向为小波分析, 分形几何和人工神经网络。

Fast Fractal Image Coding

He Aijun, Ma Zhengming

(Information Processing Laboratory, Department of Radioelectronics, Zhongshan University, Guangzhou 510275)

Abstract This paper presents a new method for fast fractal image coding. This method speeds up the coding process by adaptive partition of image, moment classification of domain pool, orthogonal decomposition of domain blocks and range blocks, as well as dynamic search by pixel after pixel. The experimental results show that the method presented in this paper performs well in speeding up the coding process, especially for those images that have much more details in them.

Keywords Fractal image coding, Adaptive image partition, Moment classification, Orthogonal projection

强强联手,共同迈向 21 世纪

北京香格里拉饭店——1999年9月9日——Diamond多媒体系统公司(股市代码:DIMD),作为一个PC多媒体和Internet连接产品的领导者,今天宣布:联想电脑公司是目前主推的面向专业用户的中高档工作站“补天”系列选择了Diamond的Fire GL1专业图形技术,Diamond与联想合作集成基于Fire GL1技术到新型的“补天2200”工作站中。

Diamond的Fire GL1技术提供了两倍带宽的128-位引擎达到快速的2D和3D图形性能,还为优化Windows NT平台结合了行业的第一个256-位图形引擎,Fire GL1技术能优化多处理器系统(具有多个OpenGL驱动)和支持Intel新款的Pentium® III线性SIMD扩充,这个技术也支持并优化最主要的CAD/CAM,数字化内容的创建,动画和虚拟化应用的软件驱动。Fire GL1技术,能定制专业化的3D性能设置和调节他们软件的应用。它具有一个优良的硬件设计和软件驱动,通过Fire GL1技术能优化到最高的性能,在今年6月份《个人电脑》杂志的3D图形卡测试中,Fire GL1以其令人惊异的高分和优异的显示效果勇夺专业产品的桂冠。

联想补天系列工作站基于Intel CPU和Windows NT Workstation操作系统,配备了先进的功能组件,具有强大的计算和处理能力、海量信息存储能力以及高速的通信能力,这次联想与Diamond公司合作的主要产品补天2200系列是联想目前主推的面向专业用户的中高档工作站。该款产品采用业界最新技术,支持最新发布的P III/600MHz CPU,主机系统性能强大,运行稳定,兼容性好,可以满足大型应用软件对系统资源的苛刻要求。预装系统监控软件,可监控温度、电压等关键参数,并能在参数失常时自动报警、预装中文NT4.0和最新的NT4补丁程序可以让系统安全度过2000年。

支持双CPU的补天2200工作站配备Diamond公司的Fire GL1(32M)专业显示卡,该产品对双CPU工作站给予了特殊支持。它可以利用一颗CPU进行几何处理运算,而另一颗CPU进行渲染运算,而对软件是不是多线程,则没有什么要求,使双CPU工作站拥有了广泛的应用价值。可以说,Diamond公司的Fire GL1显示卡为联想补天工作站注入了新的强劲动力,是两款互补性极强的产品,这种搭配使其性能都得到了充分的发挥。

此次Diamond公司与联想的合作是专业显示卡厂商与工作站厂商的强强联手,用户将从双方的合作中获益。通过此次合作,“结联世界,着想中国”的联想在引进国际先进技术,加强国内CAD/CAM/CAE的普及和应用方面迈出了实质性的一步,双方对这次联手倾注了不少信心,相信会成为21世纪一颗耀目的明星。