

基于阵列的固定块全搜索分形编码并行计算结构

赵 伟

(河南大学 国际教育学院, 河南 开封 475001)

摘 要: 与 DCT 结合的分形编码能够利用两种技术的优点, 在提供较高压缩比的同时确保图像的质量. 分形编码对于计算量的需求是制约其编解码速度进而影响其应用实时性的主要因素. 本文针对固定块大小分形编码与 DCT 结合的编码结构的计算步进行分析, 针对占其主要计算开销的固定块大小全搜索分形编码结构, 研究其数据局部性和计算并行性, 应用阵列结构, 提出一种结构规整、易于扩展并能够方便实现的并行计算结构. 该结构能够映射为 VLSI 实现, 以高效的硬件结构提供编解码计算结构所要求的实时性.

关键词: 分形编码; 固定块; 阵列结构; 并行计算

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2013)09-0164-05

Array-Based Parallel Computing Architecture of Fixed-Size-Block Full Search Fractal Coding

ZHAO Wei

(School of International Education, Henan University, Kaifeng 475001, China)

Abstract: The combination of fractal coding and DCT can take advantage of the benefits of both technologies. Higher compression ratio is provided, and at the same time, the quality of image is acceptable. The calculation amount of fractal coding is the main factor, which is restricting its codec speed, and affects its real-time applications. In this paper, fixed-size-block full search fractal coding and DCT encoding structure calculation step is analyzed. The data locality and computational parallelism of full search fractal coding structure is studied, which is the major computational overhead. A regular parallel computing structure is proposed by using array structure. This structure is scalable and can be easily mapped to VLSI implementation. The efficient hardware structure can meet the real-time requirements of the codec computing structure.

Key words: fractal encoding; fixed-size-block; array architecture; parallel computing

1 引言

为应对大数据量图像数据对于存储和传输的诸多挑战, 各种图像压缩技术得到了非常广泛的研究, 分形编码由于其高压比、任意尺度下的重构、快速编码等特点, 成为了具有广阔应用前景的技术. 1988 年 Barnsley 利用图像整体与局部的自相似性, 提出的一种应用迭代函数系统理论实现的分形图像压缩编码为该项研究奠定了基础^[1]; 1990 年 Jacquin 提出的可由计算机完全自动实现的分形图像编码算法为分形编码在图像压缩中的应用打开了实现之

路^[2]. 研究快速的分形编解码算法、提高压缩图像的质量以及研究基于分形的低码率视频编码是分形图像压缩技术的主要研究热点. 为了获得快速度编解码结构、提高压缩图像的质量, 结合传统的时域/频域信号处理方法、优化编码技术出现了很多分形技术与其他技术的组合研究. 目前, 影响较为广泛的主要包括: 分形技术与小波变换结合编码、与 DCT 变换结合编码、与加权有限自动机结合编码、与向量量化结合编码、与遗传算法结合编码、与 FFT 算法结合编码、与非线性模型结合编码、与算术结合编码等^[3,5,6,7,8,9]. 这些技术的研究在提高压缩比并保证

图像质量方面已经显示出明显的优势。

随着集成电路工艺技术的发展,单芯片所能集成的电路规模为数字信号处理的硬件加速结构提供了更为有效的实现途径,面向计算结构的专用指令集处理器设计已经成为目前移动终端乃至嵌入式电子系统设计中加速信号处理的典型实现方法^[4]。针对分形图像编码的计算结构进行分析,设计能够适应实时性要求的编解码硬件计算结构,能够在实现的底层为分形技术的成功应用提供支持。

本文针对固定块分形与 DCT 结合的视频编码结构的计算步进行分析,针对占其主要计算开销的固定块大小全搜索分形编码结构,研究其数据局部性和计算并行性,提出一种结构规整、易于扩展并能够方便实现的并行计算结构,该结构能够映射为 VLSI 实现,以高效的硬件结构提供编解码计算结构所要求的实时性。

2 固定块分形与 DCT 结合的编码结构

2.1 分形图像编码的基本算法

分形图像编码基于图像中存在的自相似结构,利用一组收敛的仿射变换来重建与原图像高度逼近的吸引子图像,典型流程如下^[2,3,5,8]:

- (1) 根据输入图像产生定义域子块 D 和值域子块 R ;
- (2) 如非固定块,则将定义域子块收缩至与值域子块 R 同尺寸;
- (3) 采用 8 种对称变换对定义域子块进行扩充,包括: 0° 、 90° 、 180° 、 270° 旋转,关于水平中轴线、垂直中轴线、第一对角线、第二对角线对称;
- (4) 根据匹配映射函数进行最优搜索。

编码的过程就是搜索一个定义域子块从而找到值域子块的最佳逼近。一般说来,有两种搜索方法。一种是全搜索另一种是快速搜索。全搜索算法将所有的定义域子块与值域子块进行比较,而快速搜索算法只用某些特殊的定义域子块与值域子块比较。全搜索算法与快速搜索算法相比,具有更好的性能,并且拥有非常规律的数据流。

2.2 结合 DCT 的固定块分形编码结构

离散余弦变换 (Discrete Cosine Transformation, DCT) 具有能量紧凑的特点,其与分形编码的结合能够利用二者各自的优点。在图像分块之后,对适合于分形编码的区块采用自仿射分形结构编码方法以保持高压缩率;对不适合于分形编码、匹配误差较大的区块采用 DCT 变换编码方法,以保持解码

图像质量。文献[6]针对 QCIF (Quarter Common Intermediate Format) 图像提出了一种结合 DCT 的固定块大小全搜索分形 (Fixed-Block-size Full-search Fractal, FFF) 编码方案,如图 1 所示。

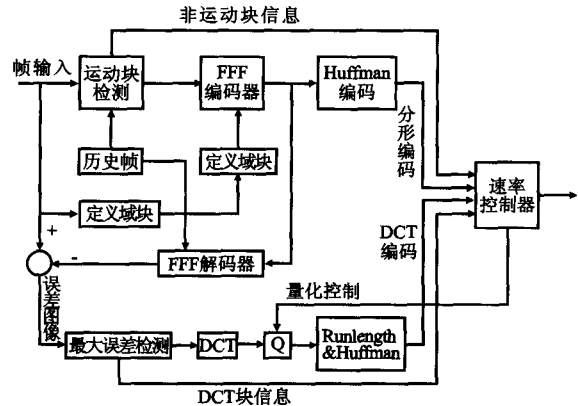


图 1 结合 DCT 的固定块分形编码结构

其中,固定块大小全搜索分形编码器对“平移”部分进行编码。将所得的压缩映射进一步用 Huffman 编码进行无损压缩,并且也发送到解码器中。同时,将所得的压缩映射用固定块大小全搜索分形解码器进行局部解码。从原始图像中减去分形解码图像得到的是残差图像,然后残差图像进行 DCT 变换,量化、runlength 编码与 Huffman 编码。图 1 中的所有模块中,固定块大小全搜索分形编码起着至关重要的作用。仅固定块大小全搜索分形编码的计算量就是总计算量的约 95%,提高固定块大小全搜索分形编码器的计算效率是该结构能够有效应用的关键因素。

3 固定块大小全搜索分形编码器的计算结构分析

采用均方根对定义域子块与值域子块进行比较的固定块大小全搜索分形计算过程如下^[3,5,8]:

- (1) 针输入图像,确定两个包含 $N \times N$ 个像素的子块, $d_1, \dots, d_{N^2}$ (定义域子块 D), $r_1, \dots, r_{N^2}$ (值域子块 R);
- (2) 根据变换产生对定义域子块进行扩充,定义旋转操作为 $r(D, x)$, 翻转操作为 $f(D)$, 则得到 8 个等距值 $r(D, 0)$ 、 $r(D, 90)$ 、 $r(D, 180)$ 、 $r(D, 270)$ 、 $r(f(D), 0)$ 、 $r(f(D), 90)$ 、 $r(f(D), 180)$ 、 $r(f(D), 270)$;
- (3) 寻找比例因子 s 和偏移量 o 使 E 值最小。

$$E = \sum_{i=j=1}^N (s \times d_{ij} + o - r_{ij})^2 \quad (1)$$

该处理过程是一个寻求 E 最小值的过程,即当

s 与 o 的偏导数是零时, E 达到最小值. 因此, 对公式

(1) 分别对 s 和 o 求偏导数, 计算得到:

$$s = \frac{[N^2 \sum_{i=j=1}^N d_{ij} r_{ij} - \sum_{i=j=1}^N d_{ij} \sum_{i=j=1}^N r_{ij}]}{N^2 \sum_{i=j=1}^N d_{ij}^2 - (\sum_{i=j=1}^N d_{ij})^2} \quad (2)$$

$$o = \frac{1}{N^2} [\sum_{i=j=1}^N r_{ij} - s \sum_{i=j=1}^N d_{ij}] \quad (3)$$

$$E = \frac{1}{N^2} [\sum_{i=j=1}^N r_{ij}^2 + s(s \sum_{i=j=1}^N d_{ij}^2 - 2 \sum_{i=j=1}^N d_{ij} r_{ij} + 2o \sum_{i=j=1}^N d_{ij}) + o(oN - 2 \sum_{i=j=1}^N r_{ij})] \quad (4)$$

这样编码过程则转换为针对定义域块和值域块的像素计算问题. 其中, 对于所有的定义域子块, 值域子块中的 $\sum_{i=j=1}^N r_{ij}^2$ 和 $\sum_{i=j=1}^N r_{ij}$ 项是固定的, 对于所有的值域子块, 定义域子块中的 $\sum_{i=j=1}^N d_{ij}^2$ 和 $\sum_{i=j=1}^N d_{ij}$ 是固定的. 表 1 给出了各项的计算量.

表 1 各多项式的计算量

多项式	乘法计算量	加法计算量
$\sum_{i=j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$	N^3	$N^3 - N$
$\sum_{i=j=1}^N d_{ij}^2 \Rightarrow \sum D \cdot D$	N^2	$N^2 - 1$
$\sum_{i=j=1}^N r_{ij}^2 \Rightarrow \sum R \cdot R$	N^2	$N^2 - 1$
$\sum_{i=j=1}^N d_{ij} \Rightarrow \sum D$	/	$N^2 - 1$
$\sum_{i=j=1}^N r_{ij} \Rightarrow \sum R$	/	$N^2 - 1$

对于 DCIF 格式的 176×144 大小的图像, 行列均按照 2:1 抽取后得到 88×72 的图像, 以固定块大小 8×8 进行全搜索, 则每一个值域子块与 5120 个定义域子块比较, 参考表 1 给出的计算量, 进行公式(2)~(4)的计算量之庞大是显而易见的.

4 固定块大小全搜索分形编码器的并行计算结构

虽然固定块大小全搜索分形编码要求非常庞大的计算量, 但是其针对块图像数据的矩阵计算结构的数据结构规整性、数据级并行以及时空局部性, 对于基于阵列结构的并行处理具有良好的适应性^[9-11], 本节在进行基于数据局部性的基础上进行计算结构的归一化, 在此基础上, 提出应用阵列化的数据存储结构和脉动阵列处理结构来进行固定块大小全搜索分形编码器的并行计算, 并针对 VLSI 的

实现开销, 提出面向性能的扩展方法.

4.1 基于数据局部性的计算结构归一化

由于对于每一值域而言, 其要针对 8 个等距值定义域子块进行计算; 并且就编码过程的时间性而言, 值域子块的时间局部性持续的时间更长. 为了充分利用数据局部性, 采用固定定义域子块, 调整值域子块的方法对针对各等距值的计算中的 $\sum_{i=j=1}^N d_{ij} r_{ij}$ 进行计算结构归一化, 转换结果如表 2 所示.

表 2 面向值域子块调整的归一化计算结构

等距值	原计算结构	面向 R 的归一化计算结构
$r(D, 0)$	$\sum r(D, 0) \cdot R$	$\sum D \cdot R$
$r(D, 90)$	$\sum r(D, 90) \cdot R$	$\sum D \cdot r(R, -90)$
$r(D, 180)$	$\sum r(D, 180) \cdot R$	$\sum D \cdot r(R, 180)$
$r(D, 270)$	$\sum r(D, 270) \cdot R$	$\sum D \cdot r(R, 90)$
$r(f(D), 0)$	$\sum r(f(D), 0) \cdot R$	$\sum D \cdot (r(R, 90))'$
$r(f(D), 90)$	$\sum r(f(D), 90) \cdot R$	$\sum D \cdot R'$
$r(f(D), 180)$	$\sum r(f(D), 180) \cdot R$	$\sum D \cdot (r(R, -90))'$
$r(f(D), 270)$	$\sum r(f(D), 270) \cdot R$	$\sum D \cdot (r(R, 180))'$

4.2 面向阵列的多项式计算结构分解

就需求的计算来讲, 其中多项式 $\sum_{i=j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$ 所要求的计算量最大, 计算类型最全, 以之来进行面向阵列的多项式计算结构分解, 其它各项基于其基础模型, 分解计算步骤.

4.2.1 $\sum_{i=j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$

多项式 $\sum_{i=j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$ 的计算, 包括了乘法和加法, 应用 $N \times N$ 规模的乘法计算阵列计算其中的乘法: ① 将矩阵 D 中的元素按照与乘法计算阵列相对应的位置分配为各个计算阵列中乘法器被乘数, 将 R 中的第一列按列序一次作为每一行中对应位置的乘数; ② 遍历 R 中所有列, 则可以得到所有的乘法计算结果.

应用 $N \times N$ 规模的加法计算阵列计算其中的加法, 由于加操作面向的操作数为每一次乘法运算所得到的结果, 所以对每一乘计算节点的所有 N 次乘结果进行本地累加, 并对全阵列中各个节点计算所得的本地累加和进行求和即可得到最终的多项式计算结果.

根据这样的计算分解, 阵列计算结构中每一计算节点至少包括一个乘法器和一个加法器. 同时, 乘

法器成为了面向单一计算节点的本地计算;而加法器的结果需要面向全阵列求和设计相应的操作数传递结构。

为了进行全阵列求和,对于 $N \times N (N=2^M)$ 规模全阵列求和的操作数传递结构做如下分解:

(1) 对于计算阵列中任一行 i 中的节点,其本地累加和均可作为其自身以及 $i+N/2^k$ (其中 $k=1, \dots, M$) 行对应位置的加法器输入之一;

(2) 第一次计算:对于行号小于等于 $N/2$ 的行,将其对应位置的本地累加和作为其行号加上 $N/2$ 后对应行对应位置的加法器的输入之一,同时加法器所在行的本地累加和作为加法器的另一个输入,进行半数行的求和,并将结果作为该节点的本地累加和;

(3) 第二次计算:对于行号大于 $N/2$ 且小于等于 $3N/4$ 的行,将其对应位置的本地累加和作为其行号加上 $N/4$ 后对应行对应位置的加法器的输入之一,同时加法器所在行的本地累加和作为加法器的另一个输入,进行 $1/4$ 数行的求和,并将结果作为该节点的本地累加和;

(4) 依上方法直至最后只剩一行,再针对该行中各列采用同上的列加法结构,最终得到所要求计算的结果。

4.2.2 其它各多项式的计算

与多项式 $\sum_{i,j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$ 的计算相比较,多项式 $\sum_{i,j=1}^N r_{ij}^2 \Rightarrow \sum R \cdot R$ 和 $\sum_{i,j=1}^N d_{ij}^2 \Rightarrow \sum D \cdot D$ 的计算也包括了乘法和加法两个部分,其中:

(1) 对于乘法计算而言, D 或 R 矩阵中相应位置的元素作为计算阵列中乘法器的被乘数与乘数输入,一次全阵列的乘法即可得到所有的乘操作计算结果;

(2) 对于加操作而言,其也是将各个节点的乘结果进行求和,其计算过程与多项式 $\sum_{i,j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$ 中得到各个节点本地累加和之后的计算相同。

对于 $\sum_{i,j=1}^N r_{ij} \Rightarrow \sum R$ 和 $\sum_{i,j=1}^N d_{ij} \Rightarrow \sum D$ 多项式的计算,其仅包括了求和计算,与多项式 $\sum_{i,j=1}^N d_{ij} r_{ij} \Rightarrow \sum D \cdot R$ 中得到各个节点本地累加和之后的计算相同。

4.3 计算阵列结构设计

根据 4.2 节的计算结构分析,用于加速各多项

式计算的专用计算阵列处理元的功能结构如图 2 所示。

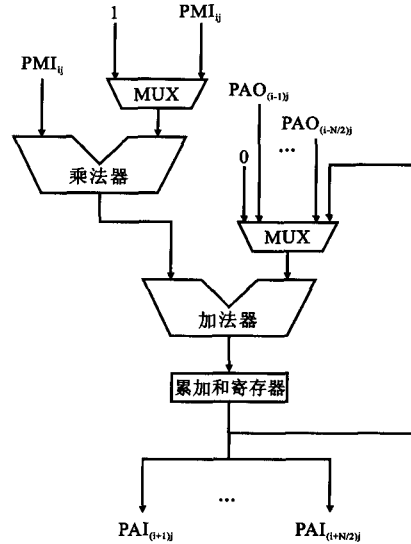


图2 计算阵列的处理元结构

每一处理元包括用于计算乘法和加法的乘法器和加法器,以及用于控制计算操作数选择的操作数 MUX、面向操作数传递的阵列间连接结构。其中 PMI_{ij} 代表第 i 行、第 j 列计算节点中乘法器的输入; PAO_{ij} 代表第 i 行、第 j 列计算节点中加法器的输出; PAI_{ij} 代表从前向行向第 i 行、第 j 列计算节点中加法器的输入传递的操作数,根据每一行在阵列中所处的位置,其前向传递的数目根据 4.2.1 节的结构定义而定。为了将计算处理元设计为一个单周期的统一结构,在乘法器中引入输入“1”的选择是为了适应多项式 $\sum_{i,j=1}^N r_{ij} \Rightarrow \sum R$ 和 $\sum_{i,j=1}^N d_{ij} \Rightarrow \sum D$ 中无乘法操作的行为;加法器输入中引入“0”的结构是为了适应各计算节点加操作中的初始累加部分。

从计算阵列的存储器访问需求来分析,最大数据带宽要求为将 D 或 R 一次全部读出并送入计算阵列对应位置,为了避免存储器访问带宽的瓶颈带来开销,将 D 与 R 的存储采用全寄存器实现,对于针对 DCIF 的固定块大小 (8×8) 全搜索分形编码器而言,其定义域和值域子块均为 64 个像素,对于 8 位灰度图像而言,采用全寄存器实现时各需要 512 个寄存器,这样的实现开销对于当前的 VLSI 实现来讲是完全可以承受的。

基于这样的设计,可以用确定的计算阵列提供所有的运算功能;应用针对不同等距值的 R 元素地址变换结构,则可以实现所有的并行计算。为了进一步提高计算阵列的利用率,在规范化的计算阵列之后再增加一行基于 4.2.1 节全阵列求和操作数传递

结构分解步骤(4)中结构的加法器,将多项式计算中最后一行的加法计算与后续计算对于阵列的使用并行化.图3给出了以固定块大小为 8×8 的分形编码所使用的阵列的结构示意图.

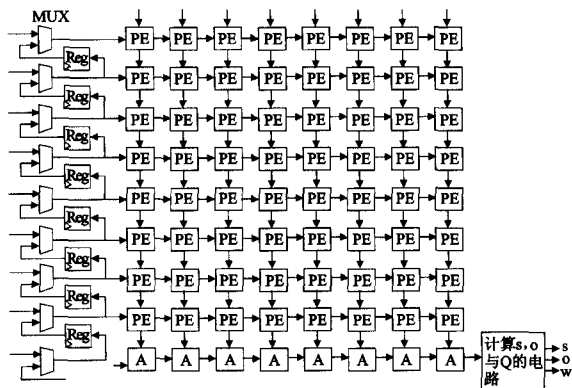


图3 针对分形编码的脉动计算阵列

根据以上的结构,针对 QCIF 格式的图片,以 8×8 的固定块大小,采用并行乘、加且均为单周期的 PE 结构,一个完整的 VLSI 实现大约需要 37 万等效门的电路,其在 120MHz 的工作主频下,即可实现 24 帧每秒速率下的一个在等距值与值域子块的计算.为了进一步加速该处理过程,可以通过 8 个脉动计算阵列的并联来实现 8 个等距值的并行处理,该并联加速仅需要根据表 2 提供的计算归一化结构为每一等距值与值域子块的脉动计算阵列设计从值域子块寄存器读取数据的结构.该结构可以通过简单的行、列索引倒序、折叠得到,在 VLSI 设计中非常直观.此时,采用寄存器实现等距值数据和值域数据存储的数据带宽优势可以得到明显的利用,全处理结构的实现开销接近 370 万等效门,在 130nm 典型 CMOS 工艺下,其性能和可实现性均可以得到保证.

5 结束语

为了加速编码速度并确保图像质量,分形编码的算法级编码加速技术、分形编码与传统时频域处理技术的结合都得到了比较广泛的研究.本文针对固定块分形与 DCT 结合的视频编码结构中占其主要计算开销的固定块大小全搜索分形编码结构,对

其计算的数据局部性和计算并行性进行分析,将编码要求的计算项映射为一阵列结构的并行计算过程,该计算结构规整、易于扩展并能够方便实现的并行,能够方便地映射为 VLSI 实现,在集成电路技术已经能够提供非常良好的可实现的情况下,以高效的硬件结构提供编解码计算结构所要求的实时性.

参考文献:

- [1] Barnsley M F. Fractal everywhere [M]. New York: Academic Press, 1988.
- [2] Jacquin A E. Image coding based on a fractal theory of contractive image transformation [J]. IEEE Tran. Image Processing. 1992, 1(1): 18-30.
- [3] Fisher Y. Fractal image compression: theory and application [M]. Berlin: Springer Verlag, 1995.
- [4] Dake Liu. Embedded DSP processor design: application specific instruction set processors [M]. Morgan Kaufmann: [s. n.], 2008.
- [5] Lazar M S, Bruton L T. Fractal block coding of digital video [J]. IEEE Trans on Circuits and Systems for Video Technology. 1994, 4(3): 297-308.
- [6] Kingwa Fu. Very low bit-rate fractal video coding by genetic algorithm [M]. Hong Kong: University of Science and Technology, 1995.
- [7] 李文敬, 黄容伟, 廖伟志. 基于相对梯度的自适应图像分形压缩并行算法 [J]. 微电子学与计算机, 2007, 24(10): 115-117.
- [8] 周一鸣, 张超, 张曾科. 基于局部方差和 DCT 变换的混合分形图像编码算法 [J]. 计算机科学, 2007, 34(10): 241-243.
- [9] 祝世平, 陈菊婧. 基于预搜索的高效双目分形视频编码 [J]. 光学精密工程, 2012, 20(3): 675-684.
- [10] John L H, David A P. Computer architecture: a quantitative approach (fifth edition) [M]. Morgan Kaufmann: [s. n.], 2011.
- [11] 梁刚, 赵伟, 时晨. 分布算术的并行计算技术研究 [J]. 微电子学与计算机, 2009, 26(5): 25-28.

作者简介:

赵伟男, (1979—), 讲师. 研究方向为图象处理、网络应用.