

视频信息压缩 域处理 图像编码 (17)

65-67

视频信息压缩域处理技术的新进展

New Advances in Compressed Domain Processing Techniques of Video

俞浩波 向辉 潘志庚 石教英

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

TM919.8
TP391.4

Abstract Massive amounts of video data are stored in computer in compressed forms. We can process them in the compressed domain. It reduced the amount of data we need to process and avoided the overhead of compressing and decompressing. In this paper, we will discuss the advances in the techniques of compressed domain processing of video data, then we will point out some new research directions in this area.

Keywords Data compression, Video data processing, Compressed domain, MPEG, JPEG

随着多媒体技术的发展,大量的视频数据被用在各种信息系统中,即使是普通电视质量的视频也差不多要求每小时100G字节的数据。一般,视频信息通常是以压缩形式存储的,对视频信息的操作必须首先将其解压缩,再进行处理,而且在大多数情况下,数据处理完以后还要重新进行压缩,这样就大大加重了系统的负担。

解决该问题的一种方法是直接地在压缩域进行处理,不仅可以减少所需处理的数据总量,避免压缩与解压缩的开销,加快处理速度,而且带来了一个附加的好处:增加了数据的局部性,可以更加有效地利用处理器的缓存,进一步提高处理速度。

压缩域处理技术是随着视频压缩的标准的建立而开始出现并发展的,目前的视频压缩标准如 MPEG, Motion JPEG, H. 261等,其压缩方式非常相似,都是首先利用 DCT(离散余弦变换)将图像变换到频域,然后利用人眼对视频信息高频分量不敏感的特性,将高频分量滤去,留下低频分量以达到压缩的目的。

本文系统地论述了视频信息压缩域处理技术的研究与进展,首先,简要介绍一下 JPEG, MPEG 压缩标准,其次,本文对视频信息的压缩域处理技术(主要是视频信息压缩域变换与视频信息压缩域检索)进行了分析与比较,最后指出一些最新的研究方向。

一、压缩标准介绍

目前的视频压缩标准有 MPEG, Motion JPEG, H. 261等,其中, Motion JPEG 是 JPEG 的扩展,它的每一帧都利用 JPEG 来压缩。而 MPEG, H. 261除了利用类似 JPEG 技术压缩以外,还有利用运动补偿技术压缩的帧,但在基本技术上大致相同,即都采用将视频帧分割成 8 * 8 像素块,并对其进行 DCT 变换。下面以 JPEG 为例,说明一下视频压缩的方法:

1. 将图像从 RGB 色彩空间转换到 YUV 色彩空间;

2. 将图像的 YUV 色彩空间的每一个分量分割成 8 * 8 像素块,并对其进行 DCT 变换,即:

$$X = SxS'$$

$$s(k, n) = \frac{c(k)}{2} \cos\left(\frac{2n+1}{16} \cdot k\pi\right) \quad k, n = 0, 1, \dots, 7$$

这里, $X = \{x(n, m)\}_{n, m=0}^7$ 是输入 8 * 8 像素块, $X = \{x(k, l)\}_{k, l=0}^7$ 是变换后输出的 DCT 系数块, $S = \{s(k, n)\}_{k, n=0}^7$ 是 DCT 变换系数矩阵,从这里我们可以看出 DCT 变换是一个线性正交变换;

3. 将利用 DCT 变换后的 8 * 8 系数块,通过 ZigZag 扫描成一个 64 位的系数矢量;

4. 将所得的系数矢量按照量化表的数值缩小;

5. 将缩小后的系数矢量化;

6. 利用 RLE(游程编码)方法对量化后的系数矢量编码;

7. 将 RLE 编码的系数矢量利用熵编码(如

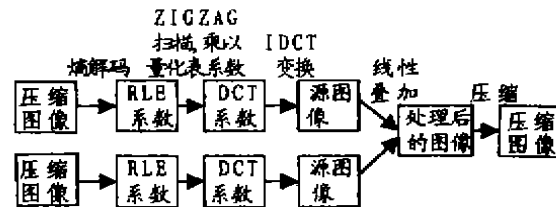
Huffman 编码或算术编码)的方法编码。

MPEG, H. 261 还有利用运动补偿技术压缩的帧(如 MPEG 中的 P, B 帧),它考虑到视频帧的时域冗余性,对于要编码的 8×8 像素块,将前面(或后面)的视频帧中的 8×8 像素块作为预测值,只存储运动矢量(预测像素块相对于实际像素块的偏移量)与预测的差值。

由于 DCT 变换是目前视频数据压缩标准 MPEG, Motion JPEG, H. 261 进行压缩变换的一个主要手段,近期的研究主要集中在 DCT 变换域的操作。而且,由于这些压缩数据流是经过熵编码的,对于它们很难直接操作,都是先将压缩数据流先解码到系数矢量或是 DCT 系数再作处理。

二、视频信息压缩域变换

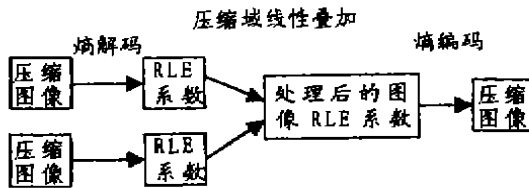
对于视频信息,我们可能希望能够进行缩放、平移、旋转以及变形等几何变换或滤波处理,以及进行视频叠加或蓝屏编辑等操作,直接地在压缩域处理无疑会大大地提高效率。有几位学者分别独立地提出了压缩域操作的有关算法。Smith 与 Rome^[1]研究了 DCT 域的一些简单操作,即对于两幅压缩图像的线性叠加,传统的方法是:



即:

$$P_{out}(i, j) = \alpha \cdot P_1(i, j) + (1 - \alpha) P_2(i, j)$$

可以在 DCT 域直接对压缩数据操作:



即:

$$DCT(P_{out}(i, j)) = \alpha \cdot DCT(P_1(i, j)) + (1 - \alpha) DCT(P_2(i, j))$$

对于两幅压缩图像的对应像素相乘,他们充分地利用了 DCT 系数的稀疏矢量数据结构来达到加速目的,算法可以用在视频的叠加、淡入淡出等效果上,实验结果显示他们的算法比传统的算法(先解压

再操作然后再压缩回去)快 10 到 50 倍,但是算法只是用到了 DCT 变换的线性特点。S. F. Chang^[2,3]推导出了 DCT 变换域的卷积定理,并利用它来计算两幅图像的对应像素逐个相乘,其速度与 Smith 的相似。

以上的操作仅仅是局部操作:输出图像像素值仅仅由输入图像的对应的像素值决定,对于全局操作,即输出图像是输入图像的许多像素的函数,则比较困难,主要因为 MPEG, JPEG, H. 261 的分块压缩的特性,而输出图像可由输入图像的许多块中的像素变换而成,Smith^[4]在 95 年提出了解决该问题的一种方法:即将压缩过程中的 DCT 变换(用张量 D 表示),ZIGZAG 扫描(用张量 Z 表示),系数矢量按照量化表的数值缩小(用张量 S 表示)结合起来,用一个张量 J 来表示:

$$J_{i,j} = \sum_{n=1}^8 \sum_{m=1}^8 S_{nm} Z_{nm} D_{i,j}$$

这样,对图像的压缩过程可以通过下面的公式来表示:

$$F_i = \sum_{j,j'} J_{j,j'} f_{i,j}$$

其中, f_i 是输入的 8×8 像素块, F_i 是输出的按照量化表的数值缩小的系数矢量。而对图像的解压缩过程可以通过下面的公式来表示:

$$f_{i,j} = \sum_{j'} J_{j',j}^{-1} F_i$$

于是,对压缩图像的线性变换可以通过下面的公式来表示:

$$H_{w,x} = J \left(\sum_{xy} T_{wxyz} (J^{-1} F_{xy}) \right) = \sum_{xy} (J T_{wxyz} J^{-1}) F_{xy}$$

其中, $J^{-1} F_{xy}$ 将系数矢量解压缩成 8×8 像素块, $T_{wxyz} (J^{-1} F_{xy})$ 对 8×8 像素块作线性变换, $\sum_{xy} T_{wxyz} (J^{-1} F_{xy})$ 对输入图像的所有 8×8 像素块作变换, $H_{w,x}$ 是变换后输出的 8×8 像素块。

这样,对于压缩图像的所有的线性操作(包括局部操作和全局操作),都可以用一个张量 $J T J^{-1}$ 来表示,并且 $J T J^{-1}$ 是预先计算好并储存在计算机中的,可以加快变换速度,但是,由于这个张量十分复杂,其数据量往往达到几十兆,如果对 $J T J^{-1}$ 不加任何处理就对压缩图像进行计算,并不比传统的算法快多少,Smith 提出了浓缩(Condensation)算法,除去 $J T J^{-1}$ 中的小系数,将这个张量近似成稀疏张量,而对变换后的图像质量并没有很大的影响。即使这样,对于一些非对称变换,如整幅图的变形,扭曲等,该算法也是不现实的。

其他学者还着眼于特殊的操作,Shen 与

Sethi^[1]提出了DCT块内变换(IBM),即通过变换 8×8 DCT块内的DCT系数的符号或重排DCT系数,实现 8×8 块内像素的多种特殊变换,而IBM与像素叠加、像素块变换相结合,就可以实现整幅图的各种特殊变换,如绕X轴、Y轴、两条对角线翻转,旋转30、60、90度等操作。Natarajan和Bhaskran^[2]提出了在压缩域中仅通过像素的平移与相加来实现整幅图的 $1/2, 1/3, 1/4$ 收缩,而且比传统的算法快4倍。所得的图像质量也十分高。对于图像任意角度的旋转,Shen与Sethi^[3]提出了基于扫描线的压缩域算法,将旋转过程分解成三次切变(shear),对每一次切变过程,分别在压缩域计算每根扫描线的偏移量,从而达到直接计算压缩域图像旋转的目的。他们的实验显示比传统的方法快2到3倍。

以上对压缩域图像进行处理的算法,可以直接地应用到Motion JPEG视频中,对于MPEG, H.261,还需考虑到那些用运动补偿算法编码的视频帧,如在MPEG中的P帧、B帧,对它们不能直接应用以上的算法,必须首先将它们转化成类似I帧格式的 8×8 DCT系数块,在这方面,S. F. Chang, Neri Merhav^[12]分别地提出了快速逆向运动补偿算法,直接地利用运动矢量和P、B帧中的误差的DCT系数,来求出P、B帧的 8×8 像素块的DCT系数。

三、视频信息压缩域检索

除了视频数据的压缩域几何变换以外,对于视频/图像信息在压缩域直接识别、检索,也取得了一些令人鼓舞的成果。特别是,目前的视频/图像信息的日益增多,而且大部分以压缩形式保存,采用压缩域技术进行视频/图像检索是非常有用的。Shen和Sethi^[12]提出了一种快速提取压缩图像的特征(如边缘信息)的算法:采用一条理想的线段来近似表示 8×8 像素块内的边缘线,利用DCT系数的特征,通过几个DCT系数的相加与相乘来计算理想的线段的位置与边缘的灰度。其速度比传统的先解压再利用Sobel算子计算边缘信息的方法提高了20倍,但是利用该算法所得的边缘线图并不连续,比较粗糙,只适用于比较简单的应用。为了得到更加精确的边缘信息,他们又发展出了基于卷积的DCT变换域边缘检测算法,利用它来得到比较精确的边缘信息的图像。该算法奠定了压缩域图像识别,基于内容的检

索的基础。

为了实现视频检索,首先要做的是压缩视频流的分割,即场景转换检测。Meng, Juan, S. F. Chang利用MPEG压缩视频流中的B帧、P帧中的向前与向后的运动矢量个数之比以及DCT系数的直流分量来快速地检测场景转换。另外,Arman, Hu, Chu, Sethi和Patel^[15]分别也在压缩域场景转换检测方面提出了各自的算法。

J. R. Smith和S. F. Chang^[16]提出了一种采用二叉树分割的基于内容的压缩域图像检索方法,他们从数据库中的图像取出具有显著的纹理特征的均质区域,每幅图像可以没有或者有多个具有显著的纹理特征的均质区域,对于给定输入的图像,纹理特征矢量从压缩域生成并且与数据库中的每幅图像的纹理特征区域比较,包含最多匹配区域的图像就是检索结果。

结论与展望 数据压缩广泛应用于多媒体、网络通讯、VOD等应用领域,然而,数据压缩也给以上应用中所涉及到的视频、图像数据进行各种的操作与处理增加了额外的开销,压缩域的视频数据的直接处理技术因而成为一项很有意义的研究领域,它避免了解压缩/再压缩等不必要的中间环节,加快了处理速度,并且由于减少所需处理的数据总量,有利于网络处理。

近期的研究主要集中在DCT变换域的操作,其中主要的原因在于DCT变换是视频数据压缩标准(JPEG、MPEG)所采用的变换技术,随着矢量量化、小波、分形等新的压缩手段的不断发展,压缩域数据处理也应考虑这些压缩算法。

目前视频压缩标准制定中并未将压缩域操作考虑在内,因此,现有的压缩域视频操作效率并未达到最优,而且操作十分复杂。这就促使我们去设计新的压缩算法,采用该算法不仅能够达到目前的视频压缩算法的效率,而且很容易使我们在压缩域对视频直接进行操作。

综合以上对视频信息压缩域处理技术的分析,我们认为,未来的研究方向主要包括这样几个方面:(1)设计新的压缩算法,支持对压缩域数据直接操作;(2)研究用小波、矢量量化、分形压缩的视频信息的压缩域处理算法;(3)设计专用的压缩域数据处理芯片。(参考文献共20篇,略)