一种从 H.264 视频流中快速计算直方图的方法

曾晓丹 周 军
(上海交通大学图像通信与信息处理研究所, 上海 200240)

摘 要 为了快速提取视频流的直方图信息,提出了一种针对 H.264 视频流的快速直方图提取算法,并首先详细阐述了该算法的基本原理与实现过程。该算法是直接利用压缩视频流中的特征信息,采用基于统计的方法来快速计算出亮度直方图。相关实验结果表明,用该算法计算出的直方图与计算解压后的图像所得到的直方图非常接近,且效率上有很大的提高,可适用于镜头分割、视频检索等相关应用领域。

关键词 直方图 H.264 特征提取 视频检索 镜头分割
中图法分类号: TN911.73 **文献标识码**: A **文章编号**: 1006-8961(2006)11-1623-04

A Fast Way to Extract Histogram From the H.264 Compressed Video Stream

ZENG Xiao-dan, ZHOU Jun
(*Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai* 200240)

Abstract In order to extract histogram information from the Compressed Video Stream rapidly, proposed a fast way to extract histogram from the H.264 Compressed Video Stream, explaine the principle of this algorithm and describe its implementing procedure in detail. This algorithm is based on the information derived from the partially-decoded video bit stream, using the statistical method to get the histogram rapidly. Experimental results show that the presented algorithm can provide satisfactory results, and the computational complexity is significantly reduced. This algorithm can be used in the region of Shot Segmentation, Video Retrieval, and etc.

Keywords histogram, H.264, feature extraction, video retrieval, shot segmentation

1 引 言

近年来,视频处理技术获得了飞快的发展。由于随着视频数据量的快速增加,对视频数据的有效管理变得很迫切,所以需要有效的检索技术。在视频检索中一般先要进行镜头分割,因为一个视频片段被分割成镜头集后,便可以用镜头中的一帧或几帧(关键帧)对相应的镜头内容进行概括性的描述。直方图方法[1~3]是镜头分割中普遍用到的方法,因为它简单,效果显著。

现有的镜头分割方法可以分为:基于压缩域的和基于非压缩域的两类。许多基于非压缩域的分割技术都是直接利用像素值来计算连续两帧的相似性。实际上,由于未压缩的视频信息的数据量非常大,绝大多数的视频序列都是经过压缩后再进行传输和存储的,因此不能采用非压缩域分割技术。基于压缩域的分割技术可以直接从编码压缩的数据中提取有用的特征信息来对镜头进行分割,如 DCT (discrete cosine transform)系数、运动矢量等,由于这样可以大大减低计算的复杂性,因此,基于压缩域的镜头分割算法是目前研究的热点,本文正是基于这一考虑,直接从压缩域来提取图像的直方图信息,以利于后期的镜头分割和视频检索等扩展应用。

基金项目:国家"863"高技术研究发展计划项目(2005AA103310)
收稿日期:2006-06-20;**改回日期**:2006-08-28

第一作者简介:曾晓丹(1981~),男。2004 年获南京邮电大学理科学士学位,现为上海交通大学硕士研究生。主要研究方向为网络多媒体。E-mail:zengxdj@sjtu.edu.cn

2 利用统计方法计算亮度直方图

由于图像亮度值非负，且只占据有限的范围[0~255]，所以图像 4×4 块亮度的均值也必在此范围内。若假设其标准差相对于此范围而言很小，则归一化的亮度就可以作为亮度概率的合理近似。由文献[4]可知，多于 98% 的块的标准差是小于 30 的，所以 4×4 块的直方图是近似服从高斯分布的，并且文献[4]中的实验表明，99.7% 的数据点是落在像素亮度的均值 $\bar{D}$ 附近 $\pm 3\sigma$ 的范围里的。本文便通过从 H.264 压缩码流中提取图像块的均值和方差来计算图像的亮度直方图。

2.1 均值的计算

由文献[5,6]知，根据残差数据类型的不同，H.264 中的变换可分为以下 3 种（如图 1 所示）：①所有 4×4 块（块 0~15、18~21、22~25）的基于 DCT 的整数变换；②4×4 亮度 DC 系数矩阵（图 1(a)的 -1 块）的哈达马变换（只用于帧内 16×16 预测模式）；③2×2 色度 DC 系数矩阵（图 1(b)、(c)的块 16、17）的哈达马变换（用于所有宏块）。本文只计算亮度直方图。

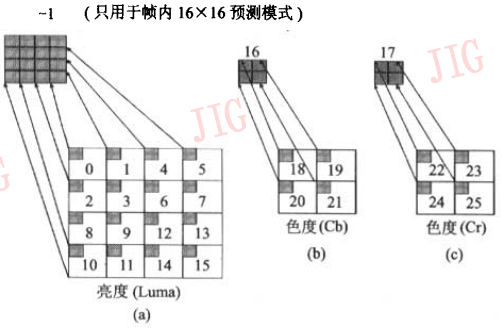


图 1 变换块的结构[6]

Fig. 1 Structure of the transform block[6]

(1)对 16×16 帧内预测的宏块进行哈达马反变换即可得到该宏块的残差的 4×4DC 系数矩阵，由式(1)可知，矩阵中的每一个 DC 系数值都对应原残差宏块中的一个 4×4 块的残差系数均值：

$$\bar{D}_r = \frac{1}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} D^r_{x,y} \tag{1}$$

其中，$\bar{D}_r$ 是 16×16 残差宏块中每个 4×4 块的残差系数均值（下角 r 代表 residual，下同），$D^r_{x,y}$ 是每个 4×4 块中相应位置上的残差系数值。接着提取该宏块的预测模式，即可得到宏块的预测值，将其分成 16 个 4×4小块，并计算每一个小块的预测系数均值：

$$\bar{D}_p = \frac{1}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} D^p_{x,y} \tag{2}$$

其中，$\bar{D}_p$ 是 16×16 预测宏块中每个 4×4 块的预测系数均值（下角 p 代表 prediction，下同），$D^p_{x,y}$ 是每个 4×4 块中相应位置上的预测系数值。

(2)对于其他预测类型的宏块：若是帧内预测（非帧内 16×16 模式），则可以直接从压缩码流中提取出残差块的残差系数均值 $\bar{D}_r$，同样的根据其预测模式，便可以得到预测块的预测系数均值 $\bar{D}_p$；若是帧间预测，则计算就相对比较繁琐，因为其涉及到运动补偿的问题，编码传输的是运动向量和运动补偿残差的变换系数，在解码重建图像时，还要通过运动向量信息来获得参考帧中对应区域的预测系数值，然后才能叠加到解码的残差数据上。所以，此时的预测系数均值要通过运动向量指向的参考帧中相应块的预测系数均值的加权平均来求得，为此本文参考文献[7]中的方法，并结合实际应用进行了改进：

$$\bar{D}_p = \frac{1}{16}\sum_{r\in A} N_r \bar{D}_r \tag{3}$$

其中，$\bar{D}_p$ 是 4×4 块 $\boldsymbol{p}_{ref}$ 的预测系数均值，$\bar{D}_r$ 是块 $\boldsymbol{r}$（块 $\boldsymbol{r}$ 指图 2 中的 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3,\boldsymbol{p}_4$）的预测系数均值，$N_r$ 是块 $\boldsymbol{r}$ 中被参考块 $\boldsymbol{P}_{ref}$“覆盖”的像素的数目，$\boldsymbol{A}$ 是被参考块覆盖的块的集合（$\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3,\boldsymbol{p}_4$）。上面的表达式只是一个近似值，但在实际的操作中，由于被省略的部分非常小，故通常情况下可忽略不计。$\bar{D}_p$ 可作为当前块（current block）的预测系数均值，当前块的残差系数均值可直接从压缩码流中提取。

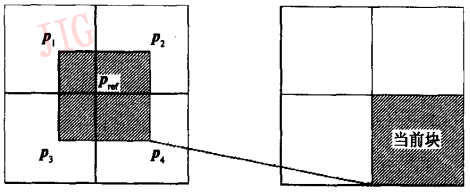


图 2 参考块，运动向量和当前块[7]

Fig. 2 Reference block, MV and current block

2.2 方差的计算

各 4×4 块中像素值的方差可用下式计算：

$$\sigma^2 = \frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3} D^2_{x,y} - \bar{D}^2 \tag{4}$$

其中，$D_{x,y}$ 是 4×4 块中相应位置上的原始像素值，$\bar{D}$

是该 4×4 块的原始像素值的均值,由文献[4]可知,式(4)的计算可近似为

$$\sigma^2 = \frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3}(D^r_{x,y}+\bar{D}_p)^2 - (\bar{D}_r+\bar{D}_p)^2$$
$$= \frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3}D^{r2}_{x,y} - \bar{D}_r^2 \quad (5)$$

其中,$\bar{D}_r$ 是每个 4×4 块的残差系数均值,$\bar{D}_p$ 是该 4×4块的预测系数均值,$D^r_{x,y}$是块中每个相应位置上的残差系数值。

因为在正交变换中不会有能量损失,故有

$$\frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3}D^{r2}_{x,y} - \bar{D}_r^2 = \frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3}D^{t2}_{u,v} - \bar{D}_r^2 \quad (6)$$

也就是说,4×4 块的方差可以从变换域直接得到,即

$$\sigma^2 = \frac{1}{4^2}\sum_{x=0}^{3}\sum_{y=0}^{3}D^{t2}_{u,v} - \bar{D}_r^2 \quad (7)$$

其中,$D^r_{u,v}$是变换域中相应位置上的残差系数值。

2.3　直方图的计算

由文献[4]中的统计结论知道,图像中各 4×4 块的直方图是近似服从高斯分布的,令块 b 在亮度值 i 处的高斯函数的值为 $G_{b,i}$,即

$$G_{b,i} = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\left(\frac{(\bar{D}_p+\bar{D}_r)-i}{2\sigma^2}\right)^2\right\} \quad (8)$$

整个图像亮度值的直方图为

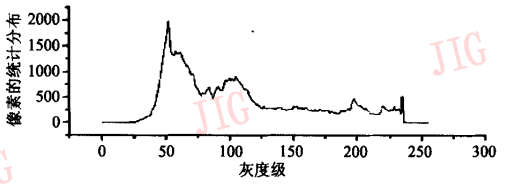
$$H_i = \sum_{\forall b\in M}H_{b,i}$$
$$= \sum_{\forall b\in M}\frac{16}{\sqrt{2\pi}\sigma_b}\exp\left\{-\left(\frac{(\bar{D}^p_b+\bar{D}^r_b)-i}{2\sigma_b^2}\right)^2\right\} \quad (9)$$

其中,M 指整个图像中的 4×4 块,其每个小块的预测系数均值、残差系数均值和方差分别为 $\bar{D}^p_b$、$\bar{D}^r_b$、σ_b^2,$i\in[0,255]$。

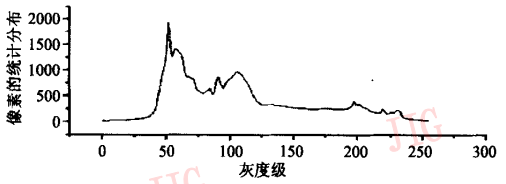
一般传统的方法先将压缩的视频图像解压,然后计算图像的像素分布直方图,做完相应的处理后再进行压缩,这样由于运算的时间复杂度和空间复杂度都很大,因而不利于实际的应用。本文所采用的方法是直接在压缩域进行直方图统计,从空间复杂度来说,由于其不需要存储解压后的图像,从时间复杂度来说,由于其不需要进行耗费时间的 IDCT 和 DCT 等运算,因此可快速地进行计算。

3　实　验

为了验证本文方法效果,选取不同类型的图像序列来作为实验素材进行了实验,实验时,首先从 H.264 Baseline Profile 的码流中任意选取一帧,然后用两种不同方法计算其直方图,并进行比较,比较的结果如图 3 ~ 图 6 所示。



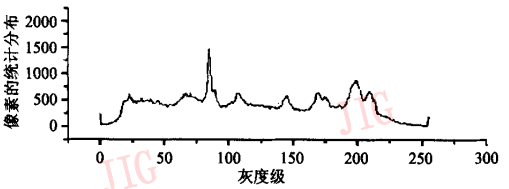
(a)



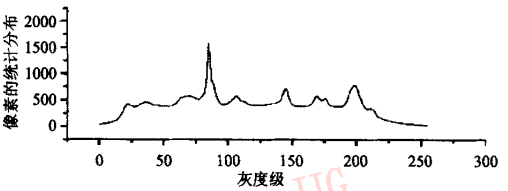
(b)

图 3　Paris. cif(I 帧)

Fig. 3　Paris. cif(I frame)



(a)



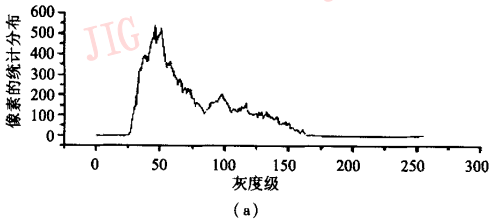
(b)

图 4　Mobile. cif (P 帧)

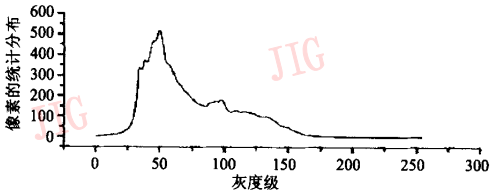
Fig. 4　Mobile. cif(P frame)

图 3 ~ 图 6 中的(a)图曲线是通过计算完全解码后的原始 YUV 图像像素值分布得到的直方图,(b)图曲线是通过从部分解码的压缩码流中提取均值和方差,然后使用高斯函数计算得到的直方图。从图 3 ~ 图 6 可以看出,两条曲线非常接近,但是(b)图曲线要平滑,因为它是从统计上来考察图像像素值的分布情况,并考虑了图像的平滑性和粗糙性,故可以减少相机和物体等运动带来的误判。

对于试验的效率,本文也对处理流程中所涉及



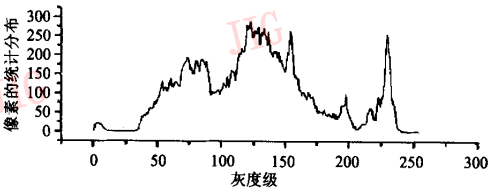
(a)



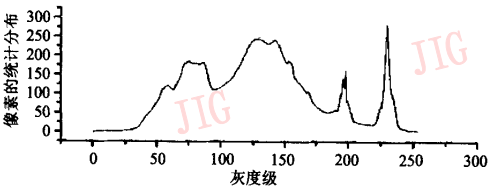
(b)

图 5 Salesman. qcif(I 帧)

Fig. 5 Salesman. qcif(I frame)



(a)



(b)

图 6 Silent. qcif(P 帧)

Fig. 6 Silent. qcif(P frame)

到的各模块进行了较为详细的分析:

如果先不考虑解码处理后的再编码过程,而仅考虑解码分析中两种方法所存在差别的部分,那么对于每个 4×4 块,直接计算需要 112 次累加运算,128 次乘法运算和 16 次比较运算;而统计方法则需要 16 次加法运算、16 次乘法运算和一次指数运算,而且通过表 1 和表 2 可以看出,在编解码中,IDCT 和 DCT 所占用的时间比例非常大。通过上面的分析可知,运用统计的方法直接从压缩域来计算直方图在效率上有了很大的提高。

本文在计算中有以下两处做了近似处理:一是计算帧间预测的预测系数均值;二是计算 4×4 块中像素值的方差。结合文献[4,7]的结果以及本实验显示的结果,可以表明该近似处理是合理的。

表 1 H.264 编码器模块性能

Tab. 1 Module performance of H.264 coder

各单元名称	各单元所占时间比例(%)
模式选择(包含插值)	53.6
变换、量化	8.9
反量化、反变换	9.2
编码(UVLC)	4.3
去块效应滤波	13
其他	11

表 2 H.264 解码器模块性能

Tab. 2 Module performance of H.264 decoder

各单元名称	各单元所占时间比例(%)
反 DCT 变换(IDCT)	40.8
逆量化、逆扫描和逆预测	24
数据分析和变长解码	13.2
其他	22

4 结 论

现在大量存储和传输的视频数据都是经过压缩的,如果将压缩的视频数据完全解码后再进行处理,这就要经过解压、分析处理、再压缩的过程,效率低下。本文提出的方法是直接从 H.264 压缩码流中计算亮度直方图,效果相比与直接从非压缩域处理非常接近,但是效率上却有了极大的提高。

参考文献(References)

1 Rasheed Zeeshan, Shah Mubarak. Detection and representation of scenes in videos [J]. IEEE Transactions on Multimedia, 2005, 7(6):1097~1105.

2 Haering N. A framework for the design of event detections[D]. Ph. D. dissertation, School Computer Science, University Central Gainesville, Florida, USA. 1999.

3 Zhang H J, Kankanhalli A, Smoliar S W. Automatic partitioning of full-motion video[J]. Multimedia Systems, 1993, 1(1):10~28.

4 Patel N V, Sethi I K. Compressed video processing for cut detection [J]. IEE Proceedings Vision Image Signal Process, 1996, **143**(5): 315~323.

5 ITU-T Rec. Advanced video coding[S]. Final Committee Draft, H.264/ISO/IEC 14496-10, Document JVT-F100, December 2002.

6 Iain E G Richardson. H.264 and MPEG-4 Video Compression[M]. The Robert Gordon University, Aberdeen, UK: John Wiley&Sons.

7 Yeo Boon-Lock, Liu Bede. Rapid scene analysis on compressed video [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1995, 5(6):533~544.