H.264 兼容的全景视频编码方法

熊 聪 余松煜 宋 利 杨小康
(上海交通大学电子工程系图像通信与信息处理研究所,上海 200240)
(上海交通大学上海市数字媒体处理与传输重点实验室,上海 200240)

摘 要 在拍摄全景视频时,摄像头往往仅在水平方向上移动。由于在水平移动中偶尔有轻微的上下抖动,以及传感器采集引入的噪声,直接对其进行 H.264 编码,编码效率不高。针对全景视频拍摄和 H.264 编码的特点,提出了一种对全景视频高效的 H.264 编码方法,首先将原始全景视频合成一张全景图;然后再把全景图直接转换为标准 H.264 码流。实验结果表明,采用该方法对其编码,与直接编码原始视频相比,编码效率大大提高。

关键词 全景视频 全景图 H.264 视频编码 运动矢量
中图法分类号:TN919.81 **文献标识码**:A **文章编号**:1006-8961(2007)10-1832-05

A H.264 Compatible Encoding Scheme for Panoramic Video

XIONG Cong, YU Song-yu, SONG Li, YANG Xiao-kang
(*Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai* 200240)
(*Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai Jiaotong University, Shanghai* 200240)

Abstract When capturing panoramic scene, the camera usually moves only in the horizontal direction. If we encode the captured panoramic video directly, the encoding efficiency will be low dueto the possible light dithering of the camera in the vertical direction and noise introduced by sensor. This paper presents a highly efficient algorithm for encoding panoramic video with H.264 standard according to its special characteristics. First, we convert the original panoramic video to a panoramic image. Then we can convert the panoramic image to H.264 bit-stream directly. Compared to the encoding of the original panoramic video, the algorithm we present highly improve the encoding efficiency.

Keywords panoramic video, panoramic image, H.264, video coding, motion vector

1 引 言

2003 年,活动图像专家组织(MPEG)和视频编码专家组织(VCEG)联合推出了新一代的视频压缩标准 H.264/AVC。该视频压缩标准较之以前的视频压缩标准编码效率有很大的提升。但是,对于一些特殊的场景,如摇动拍摄的静止场景的全景视频,直接用 H.264 对其进行压缩编码,效率往往不高。

这类全景视频有其独特的特点:摄像机全局运动通常可以认为是平动模型;相邻帧之间存在较多的重叠图像信息;拍摄的场景中几乎没有运动物体;拍摄中常常伴有轻微的上下抖动,从而引入运动噪声。针对上述特性,已经提出了一些特殊的编码方法来改善编码效率。MPEG-4 标准[1]中引入了基于 Sprite 的编码方法,将通过在编码端估计全局运动参数模型,然后通过全局运动补偿的方法获得预测值。Patrick 等人根据视频帧之间的纹理和运动特性,提出了基于纹理分析与合成的编码方法[2],也采用了全局运动补偿的方法来获得重建。

尽管上述方法具有较高的编码性能,如要集成到现有的 H.264 混合闭环框架中,需要对编解码器

基金项目:国家自然科学基金项目(60625103);国家高技术研究发展计划 863 项目(2006AA01Z124)
收稿日期:2007-07-05;改回日期:2007-07-24
第一作者简介:熊聪(1983~),男。现为上海交通大学通信与信息系统专业硕士研究生。主要研究方向为视频编码与传输。E-mail:congxiong@hotmail.com

进行修改。因此用标准的 H.264 解码器是无法解码的,或者说与标准不兼容。

针对上述问题,本文提出的一种新的全景视频编码算法,首先将全景视频合成全景图,去掉了视频中的上下抖动和噪声;然后将全景图通过关键帧分割获得 I、P 帧,通过中间帧插入获得 B 帧,从而将原始视频"间接地"编码为标准 H.264 码流。转换后的码流完全兼容 H.264 标准,可用 VLC(videolan client)和 MPC(media player classic)等支持 H.264 解码的播放器播放。

采用本文提出的算法对全景视频编码,解码出来的视频更加清晰,不但去掉了噪声和抖动,而且编码的码率还大大降低。编码后码流的大小,大约降为直接对其编码后的 1/10 左右,甚至小于用 JPEG 压缩的合成全景图的大小。

2　算法的提出

全景视频的相邻帧之间存在大量的像素重叠,具体如图 1 所示。当前帧内 S_1 部分像素和前一帧内 R_1 部分像素完全相同;当前帧内 S_2 部分像素,与后一帧内 T_1 部分像素完全相同。这两部分像素相对于相邻帧内对应部分的像素,仅仅是在位置上有水平方向上的相对移动。其中,S_2 部分像素的宽度(称为插入精度),是由摄像头拍摄时水平方向上的移动速度决定的。

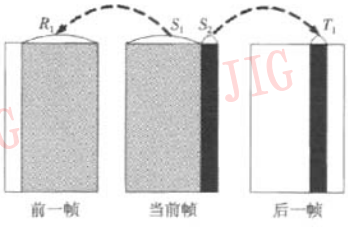


图 1　全景视频中相邻帧间的像素对应关系

Fig. 1　Co-located pixel position of neighbor frames in panoramic video

从全景视频中相邻帧之间像素的对应关系,不难看出:全景视频所表达的内容其实就是一幅全景图的内容。在编码全景视频时,为了最大限度提高编码效率,对表达相同内容的像素只需编码一次;即只需编码全景视频合成的全景图内的像素,而其余视频帧的像素参考全景图内的像素。

本文提出对全景视频进行 H.264 编码的算法,如图 2 所示,主要包括全景视频到全景图的合成、分割,编码关键帧以及插入中间帧。为方便算法的描述,文中假设摄像头拍摄时是水平从左向右移动。对于从右向左的移动的情况,处理方法类似。

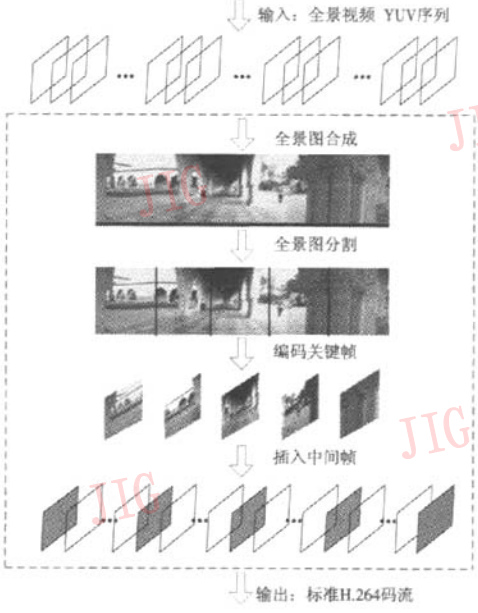


图 2　编码全景视频算法的框图

Fig. 2　Scheme of the algorithm of encoding panoramic video

2.1　全景图的合成

关于全景视频到全景图的合成,已经提出一些性能良好的算法。Peleg 等人提出了基于图像帧像素空间对齐,通过图像帧的剪切和拼接来合成全景图的算法[3]。Wexler 等人考虑了图像帧拍摄的时间因素,提出了基于空间和时间,通过全局最优匹配来合成全景图的算法[4]。针对拍摄静止场景的全景视频,利用上述的算法都能合成效果很好的全景图。

本文直接利用 Wexler 等人提出的基于空时合成算法,将一段分辨率为 $W_v \times H_v$ 的全景视频序列合成一幅分辨率为 $W_m \times H_m$ 的全景图(W_v 为视频宽度,H_v 为视频高度,W_m 为全景图宽度,H_m 为全景图高度)。合成后的全景图内,所有的像素在水平方向上对齐,去掉了拍摄中引入的抖动噪声。另外,由于全景图的高度 H_m 可能大于原始全景视频的高度 H_v(如图 6 所示),需要进行切割使得 $H_m = H_v$。

2.2 编码关键帧

首先,将全景图依次分割为多帧宽度为 W_s 的图像。如果 W_m 不是 W_s 的整数倍,由图像的后端按相反方向切割,得到最后一帧图像(该帧图像和分割的相邻帧有部分像素重叠)。

然后,将分割所得图像,作为每个 GOP(group of pictures)的关键帧进行编码。这里,GOP 分两种情况:分割全景图时,如果 W_m 不是 W_s 的整数倍,称分割所得的最后一帧图像所在的 GOP 为非完整 GOP;对于其余情况下的 GOP,统称为完整 GOP。

对完整 GOP 内的关键帧,采用 I 帧编码,同常规编码相同,需进行模式选择。

对非完整 GOP 内的关键帧,采用 P 帧编码,以进一步提高编码效率。对该帧内和相邻关键帧内重叠部分像素,采用 Inter 宏块编码;剩下部分的像素,全部采用 Intra 宏块编码。对 Inter 宏块,选择 P_L0 或 P_SKIP 模式编码;选择 P_L0 时,运动矢量由式(1)获得。

$$\begin{cases} V_x = 4 \times (W_m \bmod W_s) \\ V_y = 0 \end{cases} \tag{1}$$

2.3 插入中间帧

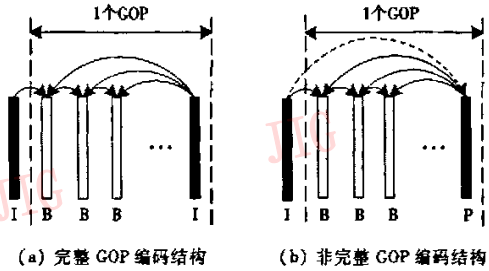
编码完当前关键帧后,还需要在相邻的关键帧间(即当前 GOP 内)插入大量中间帧。当前 GOP 内插入中间帧的个数 N_B 由 W_s,插入精度 W_{new} 以及 GOP 类型共同决定。对于完整 GOP,N_B 由式(2)获得;对于非完整的 GOP,N_B 由式(3)获得。

$$N_B = W_s / W_{new} - 1 \tag{2}$$

$$N_B = (W_m \bmod W_s) / W_{new} - 1 \tag{3}$$

所谓插入精度 W_{new},是指当前插入帧相对于前一帧,出现的部分"新"像素的宽度,如图 1 中 S_2 部分像素的宽度。考虑到 H.264 标准中参考帧的最小单位为 8×8 块,插入精度一般取值为 8 的整数倍。

由于插入的中间帧内的像素均可以在相邻前后帧内参考到完全对应的像素,所以全部采用 B 帧编码。当前 GOP 内插入的第一个 B 帧使用相邻的两个关键帧作为参考帧;其余的 B 帧,使用插入的前一帧和当前 GOP 的关键帧作为参考帧。参考帧的管理涉及到 MMCO(memory management control operation),具体可参考 H.264 标准文档[5]。



(a) 完整 GOP 编码结构 (b) 非完整 GOP 编码结构

图 3 GOP 编码结构

Fig. 3 Encoding structure of a GOP

在下面的图 4 和图 5 中,显示了所有插入 B 帧内的各部分像素,在前向和后向参考帧内对应像素的位置关系。

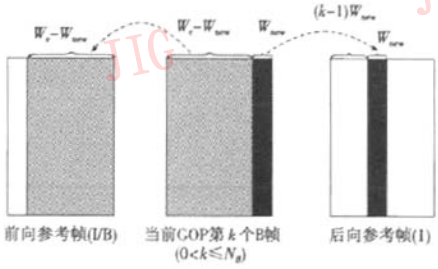


图 4 完整 GOP 中 B 帧内各部分像素的参考关系

Fig. 4 Refer position of B-frame pixels in a closed GOP

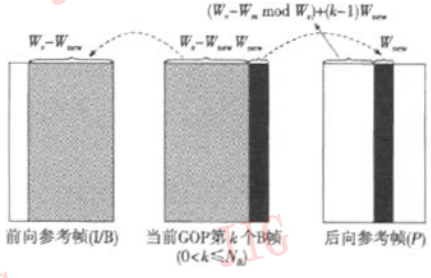


图 5 非完整 GOP 中 B 帧内各部分像素的参考关系

Fig. 5 Refer position of B-frame pixels in a unclosed GOP

如图 4 所示,对于当前完整 GOP 内插入的第 k ($0 < k \leqslant N_B, k \in \mathbf{Z}$)个 B 帧,其前向运动矢量由式(4)获得,后向运动矢量由式(5)得到。

$$\begin{cases} V_x = 4 \times W_{new} \\ V_y = 0 \end{cases} \tag{4}$$

$$\begin{cases} V_x = 4 \times (k \times W_{new} - W_s) \\ V_y = 0 \end{cases} \tag{5}$$

如图 5 所示,对于当前非完整 GOP 内插入的第 k ($0 < k \leqslant N_B, k \in \mathbf{Z}$)个 B 帧,类似地,其前向运动矢量同样由式(4)获得;后向运动矢量则由式(6)得到。

$$\begin{cases} V_x = 4 \times (k \times W_{new} - W_m \bmod W_s) \\ V_y = 0 \end{cases} \tag{6}$$

在编码每个宏块时,得到运动矢量后,宏块类型

的判定依赖于当前宏块的位置和插入精度。参考前向参考帧的宏块,可选取 B_L0_L0 或 B_SKIP 模式;参考后向参考帧的宏块,可选取 B_L1_L1 或 B_SKIP 模式。特别地,当 $W_{new}=8$ 时,需要使用 B_L0_L1 模式编码。

3　实验测试

实验是在开源代码 x264[6] rev602 版本的基础上修改的。具体测试环境为:Pentium D CPU 2.8GHz, 512M 内存, Windows XP 操作系统, Microsoft Visual Studio 2005。这里,测试了两个全景视频:其一为 60 帧 CHURCH 序列(320×240);其二为 90 帧 CAR 序列(352×288)。两者合成的全景图如图 6 所示。表 1 显示了采取固定 QP 值 26 和 CAVLC (context-based adaptive variable length coding)熵编码,对两视频进行如下两种编码方案比较。

(a) CHURCH 序列(320 × 240)合成的全景图(891 × 253)

(b) CAR 序列(352 × 288)合成的全景图(818 × 304)

图 6　由全景视频合成的全景图

Fig. 6　Panoramic image synthesized from panoramic video

第 1 种方案为直接编码全景视频。为尽量使压缩后码率最小,采用较极端的编码方式:1 个 GOP(只有第 1 帧编为 I 帧),16 个参考帧,每个 I 帧和 P 帧之间或 P 帧和 P 帧之间插入 16 个 B 帧(x264 最大支持的 B 帧个数),B 帧可作参考帧。

第 2 种方案采用文中提出的算法,插入精度取值为 8。

从表 1 中可以看到,采用第 2 种方案,编码两个序列后码流的大小分别变为采用第 1 种方案的 1/9 和 1/13 左右,甚至只有对应 JPEG 压缩的全景图(主观质量相当)的 2/3 大小。可以看出,编码效率的提高非常明显。

4　结　论

本文针对全景视频的独特特点,提出了一种非常高效的编码方法。利用该算法编码后的码流完全兼容 H.264 标准。因此,该算法可以应用于全景图

表 1 CHURCH 和 CAR 序列两种编码方案比较结果

Tab. 1 Comparison of encoding CHURCH and CAR sequences with two supplied schemes

	CHURCH 序列	CAR 序列
原始全景视频帧数	60	90
第 2 种方案编码恢复的视频帧数	72	59
第 1 种方案编码后码流大小(Byte)	229 528	276 885
第 2 种方案编码后码流大小(Byte)	24 735	20 183
第 2 种方案关键帧码流大小(Byte)	20 972	16 603
第 2 种方案中由全景视频合成的全景图(JPEG 压缩后)大小(Byte)	34 366	30 623
第 1 种方案的编码速度(fps)	26.49	22.16
全景图到 H.264 码流转换速度(fps)	307.69	252.14

像到视频的转换和常规视频编码中。

可以将该编码算法作为常规视频编码的一个增强模式。在编码时,通过全局运动估计等方法来检测待编码视频中的一段视频是否为全景视频,如果是,则可将这段视频用该模式编码。此外,在实际应用中,还需要考虑以下问题:

其一是实时性。本文中全景图的合成通过离线的方式完成。表 1 中,可以看到全景图到 H.264 码流的转换速度较常规编码快很多,然而其节省的时间是否足够前端全景图的合成待进一步验证。

其二是插入精度。虽然恢复的全景视频完全展现了原始视频的内容,但是恢复出的帧数可能与原始视频不一致。可以考虑将插入精度进一步降为 4,2 或 1,但这又会引起插入 B 帧的编码中部分宏块有残差,导致编码效率有所下降。

上述问题将在进一步工作中加以完善,以期获得编码性能和效率的提高,满足实际应用。

参考文献(References)

1 ISO. IEC 14496-2. Information Technology-Generic Coding of Audio-Visual Objects, Part 2: Visual[S].

2 Ndjiki-Nya P, Makai B, Smolic A, *et al*. Improved H.264/AVC coding using texture analysis and synthesis[A]. In: Proceedings of International Conference on Image Processing[C], Barcelona, Spain, 2003: 849~852.

3 Peleg S, Herman J. Panoramic mosaics by manifold projection[A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition[C], San Juan, Puerto Rico, 1997: 338~343.

4 Wexler Y, Simakov D. Space-time scene manifolds [A]. In: Proceedings of International Conference on Computer Vision [C], Beijing, China, 2005: 858~863.

5 ISO. IEC 14496-10, Advanced Video Coding for Generic Audiovisual Services[S].

6 x264 Source Code. http://developers.videolan.org/x264.html.