浙江理工大学学报(自然科学版),第35卷,第3期,2016年5月
Journal of Zhejiang Sci-Tech University (Natural Sciences)
Vol. 35, No. 3, May. 2016

DOI:10.3969/j.issn.1673-3851.2016.05.017

基于视频压缩域的深度图推理算法研究

冯　杰,马汉杰
(浙江理工大学信息学院,杭州 310018)

摘　要:对2D到3D视频转换过程中的深度图推理算法进行了研究。该研究以视频压缩域中的宏块为单位进行深度图推理,根据不同的宏块类型选择不同的推理策略。首先,采用基于邻块的运动估计算法对帧内宏块的运动矢量进行计算;然后,针对帧间宏块,对直接提取出的运动矢量进行滤波处理以提升其鲁棒性;最后,采用运动补偿和上采样双边滤波技术获得深度图。实验结果表明该方法可以获得平滑而可靠的深度图像,并且具有更好的深度图像质量。

关键词:2D到3D视频转换;深度图;运动矢量;压缩域
中图分类号:TP37　　**文献标志码**:A　　**文章编号**:1673-3851(2016)03-0421-06　　**引用页码**:050605

0　引　言

人类在自然界中看到的是真实的3D世界,人类所能感知到的周围环境是也是3D的,因此,能够在屏幕上再现这种真实的3D场景一直是人类梦寐以求的理想。就目前的3D发展状况而言,3D内容的极度匮乏是造成3D视频产业成长一个主要障碍。为了解决这一问题,首先需要了解人类对3D视频的感应机理。人眼的3D视觉是通过双眼间的视网膜视差来感知对象的距离即深度信息的,而目前的3D设备正是利用了人眼双目视觉系统的这一感知特点,通过分别给左右眼以对应的平面图像,根据几何成像关系,这两幅图像中不同深度位置的对象在图像间具有不同大小的视差,然后在大脑中完成视觉场景中对象间相对深度的感知重建,这样人眼就可以获得具有深度感知的立体视觉效果。为了生成3D视频,目前比较普遍的做法是采用基于深度图绘制(depth image-based rendering,DIBR)的方法来生成具有视差的左右视点图像[1]。因此如何利用原有的2D视频得到对应的深度图是2D到3D的转换领域一个非常重要的研究方向。

目前,深度图生成算法主要分成两大类:全自动算法和半自动算法。全自动算法在2D到3D转换过程中不需要人工参与,直接利用2D视频所提供的信息和深度线索估计视频帧的深度。该类方法在很多场景下可以获得比较好的深度图,但对于一些特殊的景物(如悬挂的物体等)还是不能正确获得它的深度。因此,为了获得转换速度和转换效果的平衡,目前很多研究更倾向于略有人工参与的半自动的转换方式。半自动方法首先将整段视频分成关键帧和非关键帧,并对关键帧采用人工参与方式赋予比较精确的深度值,对于非关键帧则采用深度推理方式得到相应的深度图。其中,直接对已有深度图进行推理或滤波的方法主要包括基于块的运动补偿法[2]、双向KLT跟踪算法[3]、基于块的运动估计结合三边滤波器[4]、基于扫描线的推理滤波[5]、Graph Cut与Random Walks相结合的推理[6]等。半自动方法中由于有了人工的参与可以有效的避免前期很多自动算法造成的物体前后位置误判,因而可以获得更加准确的深度图。但是目前大部分算法都是基于像素域进行的,这在实际应用中会消耗大量的计算资源。并且目前很多视频都是用压缩方式进行存

收稿日期:2016-01-16
基金项目:国家自然科学基金项目(61501402)
作者简介:冯杰(1980-),男,辽宁锦州人,讲师,主要从事视频处理方面的研究。

储的，如果在压缩域就可以对视频内容进行分析和处理可以大大提升处理速度。并且压缩视频数据都是以块为基本单位，所需要处理的数据量仅是像素域的 1/64(基于 8×8 块)或 1/16(基于 4×4 块)。因此，需要在保证深度图质量的前提下尽可能利用压缩域中隐含的信息对 2D 视频进行分析处理，就可以达到深度图生成效率与质量的平衡。

1 算法总体设计

本文算法主要针对压缩域的半自动 2D 到 3D 视频转换。在视频压缩编码过程中，I 帧一般作为关键帧，P 帧作为非关键帧。相应的，关键帧(I 帧)的深度图将采用人工的方式获得。因此，本文主要关注 P 帧图像的深度图推理算法设计。如图 1 所示，在对压缩视频进行解码后，本文的深度图推理算法主要分成 3 个主要部分：首先，对于帧内编码宏块，需要进行基于邻域的运动估计来获得它的运动矢量；其次，对于帧间编码宏块，虽然可以直接从压缩域获取运动矢量信息，但由于这些信息存在很多噪声，并不十分可靠，因此需要对其进行自适应滤波处理；最后，利用已获得的运动矢量信息、解码 2D 视频信息、关键帧或前一非关键帧的深度图信息对当前帧的深度图进行赋值和推理。

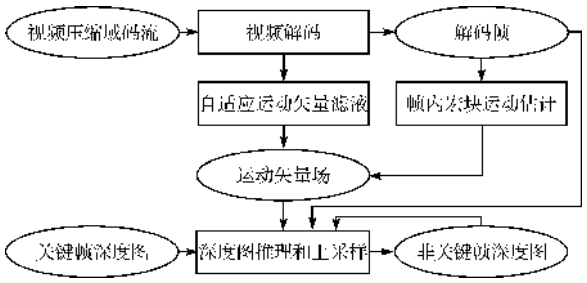

图 1 深度图推理算法总体流程

2 视频压缩域信息提取与分析

在普通的 2D 到 3D 半自动视频转换算法中，非关键帧的光流信息(即运动信息)经常被用来做深度推理。而在视频压缩域中，运动矢量是可以通过解码 P 帧或 B 帧直接获得的信息，不再需要利用像素域中的光流估计来获得运动信息，这可以极大的降低运算复杂度。视频编码器利用高效的运动估计算法获得的运动矢量刚好可以提供有效的光流信息。但是采用直接从视频码流中获取的运动矢量作为光流信息仍然存在 3 点不足。

2.1 无法提取帧内编码类型宏块的运动矢量

在视频压缩域中，P 帧中存在 3 种不同的编码宏块类型：帧内编码(INTRA)、帧间编码(INTER)和 SKIP 编码。帧间编码和 SKIP 编码类型的宏块可以利用运动估计算法在参考帧中找到一个匹配块，进而得到该类型宏块的运动矢量。但是帧内编码类型宏块主要利用周围宏块的空间插值算法来提高压缩效率，并不能从码流中获取关于该类型宏块的运动信息，因此在码流中得到的宏块运动矢量都为 0。在视频编码过程中，宏块编码模式的选择一般是按照率失真优化(rate distortion optimization，RDO)技术来实现的，这样可以获得最佳的编码效率。图 2 列举出了一帧图像的宏块编码类型信息和它的运动矢量信息。图 2 中的黑色、灰色和白色圆点分别代表采用帧内编码、帧间编码和 SKIP 编码模式的宏块，线条代表每个宏块的运动矢量。在图 2 中可以看到有很多本应处于运动状态的宏块采用了帧内编码模式，无法从压缩域的码流中获得该类型宏块的运动信息。

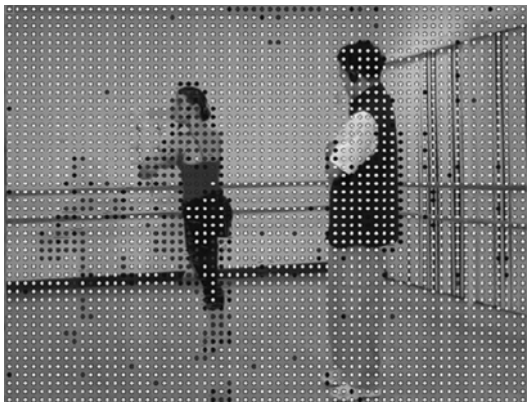

图 2 压缩帧的宏块编码类型信息和运动矢量信息

2.2 帧间宏块运动矢量存在噪声

在绝大多数情况下，视频编码所得到的运动矢量大部分是比较符合实际的运动情况的，尤其在运动对象的边界区域，在 H.264 编码标准中更是采用较小的帧间分块方式来获得更精细的块分割效果。但是在比较平坦的区域，会出现运动矢量噪声，这是因为在视频编码过程中，会采用预测运动矢量(pridict motion vector，PMV)作为计算宏块运动矢量所消耗比特数的起点。在平坦区域中，一般情况下选择预测运动矢量附近的运动矢量值会获得与采用真正运动矢量时相同的预测残差，这样会降低用来编码运动矢量的比特数，而真正的运动矢量(也许是零)并不被编码算法所采纳。因此，为了得到真实的对象运动信息，需要对帧间编码宏块的运动矢量进行滤波处理。而 SKIP 类型宏块的运动信息一般是很符合物体实际运动情况的，对于该类宏块不需进行滤波处理。

2.3　运动矢量精度不高

为了适应各种不同纹理的视频场景,视频编码过程可以采用可变块大小的运动编码方式,分块大小最大为 16×16,最小为 4×4,中间可以为 16×8/8×16/8×8/8×4/4×8 等等,因此,每个运动矢量可能对应不同的宏块大小。如果要获得统一大小的运动矢量,那么每一个大于 4×4 块的分块方式可以将其运动矢量赋值给其覆盖的 4×4 块中。但是最小的 4×4 的运动矢量精度仍然要比光流场的精度有所降低。

3　压缩域运动信息提取与深度图推理

基于上一节的分析,针对帧内宏块类型没有运动信息的问题,本文采用基于邻块信息的运动估计方法来获得帧内宏块的运动矢量,这样 P 帧中所有宏块都将具有运动信息;针对帧间宏块运动信息不准确的问题,本文采用自适应滤波算法对帧间宏块的运动矢量进行处理以提高其准确性;针对运动矢量精度不够的问题,本文用基于块的运动补偿算法推理出一个低分辨率的深度图,然后利用上采样双边滤波器来获得更加细致的深度图。

3.1　帧内宏块运动估计

在视频编码过程中,一般采用基于率失真的策略来进行模式选择,进而确定某一个宏块采用何种类型来进行编码。当帧内模式具有比帧间模式更小的率失真时,编码器将对该宏块采用帧内模式来进行编码,此时该宏块的运动矢量信息部分将不会保存,而设置为 0。因此,为了获得该类型宏块的运动信息,需要采用运动估计来预测这些帧内宏块的运动矢量。和编码过程一样,运动矢量的获得也是通过最小化解码帧中的宏块和前一帧中各块差的绝对值而得到的,如式(1)所示:

$$(mv_x^{(k)},mv_y^{(k)})=\underset{(v_x,v_y)}{\operatorname{argmin}}\sum_{(x,y)\in IMB^{(k)}}absdiff(x,y,v_x,v_y) \tag{1}$$

其中:$IMB^{(k)}$ 为第 k 个帧内宏块,$(mv_x^{(k)},mv_y^{(k)})$ 是第 k 个帧内宏块计算得到的运动矢量。绝对差值函数 $absdiff(x,y,v_x,v_y)$ 由式(2) 所定义:

$$absdiff \equiv |dec^{(t)}(x,y)-dec^{(t-1)}(x+v_x,y+v_y)| \tag{2}$$

其中:$dec^{(t)}$ 为 t 时刻的解码帧亮度分量,v_x 和 v_y 是在一定水平和垂直预设搜索范围的偏置。如果该帧中存在过多的帧内编码宏块,那么运算时间将会由于运动搜索的复杂性而大大增加。为了减少帧内宏块的运动估计计算量,采用式(3) 来决定是否对帧内宏块采用运动估计策略。

$$MEFlag^{(k)}=\begin{cases}1 & NeighbourMV_{max}^{(k)}>TH\\ 0 & otherwise\end{cases} \tag{3}$$

其中:$MEFlag^{(k)}$ 代表第 k 个帧内编码宏块是否需要进行运动估计。$NeighbourMV_{max}^{(k)}$ 是该帧内宏块周围宏块中运动矢量的最大值。TH 是一个预先设定的阈值,可以采用该帧所有运动矢量的中值来确定。

这样,对帧内宏块进行运动估计的比例可以大大降低,可以有效节省计算资源。

3.2　自适应运动矢量滤波

通过第 2 节的观察,并非所有的运动矢量都代表宏块真正的运动信息。为了得到更可靠而准确的运动信息,首先,通过基于块能量的零运动矢量判别机制来消除平滑区域内的不可靠运动矢量。帧间编码宏块经过熵解码后得到残差块的 DCT 系数信息,利用该信息可以获得每个 4×4 残差块的能量大小 $En_{4\times4}(i,j)$,该能量值可以通过式(4) 进行计算:

$$En_{4\times4}(i,j)=|DC(i,j)|+|AC_{01}(i,j)|+|AC_{10}(i,j)|+|AC_{11}(i,j)| \tag{4}$$

其中:$DC(i,j)$,$AC_{01}(i,j)$,$AC_{10}(i,j)$,$AC_{11}(i,j)$ 分别是第(i,j) 个残差块的左上角 DCT 系数。DC 系数是 DCT 变换中最重要的系数,它可以代表整个残差块的总体平均能量值。其他 3 个系数 AC_{01}、AC_{10}、AC_{11} 分别代表残差块水平分量、垂直分量和对角线分量的能量值。这 4 个系数可以代表 4×4 残差块的能量大小。当该块处于平滑区域时,该能量通常为 0。

接下来,采用两步滤波处理来对每个块的运动矢量进行滤波。第一步将每一个原始的运动矢量按照从上到下从左到右的顺序进行滤波处理,当该块满足以下 3 个条件时,该块的运动矢量将被置为 0:

条件 1:该块的能量大小 $En_{4\times4}(i,j)$ 为 0;

条件 2:该块左边块的运动矢量为 0;

条件 3:该块的运动矢量值小于一定阈值。该阈值的设定有两种方式,一种采用该帧中所有块运动矢量的平均值,另一种采用用户自定义的值。

第二步仍然采用与第一步相似的方法进行滤波处理,区别在于处理的顺序变为从下到上从右到左。相应的条件 2 换为该块右边块的运动矢量为 0。通过这两步滤波可以将大部分平滑区域的运动矢量置为 0,大大提高了后续处理的准确度。

在目前最流行的视频编码器 H.264 编码过程中,具有 8×8/8×4/4×8/4×4 这种小分块模式的

块运动矢量通常比 16×16/16×8/8×16 这种大宏块的运动矢量更加可靠，所以将针对这两种不同分块类型的运动矢量采用不同系数的二维 3×3 均值滤波器来得到更加可靠的运动信息。该滤波器的详细细节可以参考文献[7]。

图 3 显示了一帧未经过和经过帧内运动矢量估计与运动矢量滤波的运动矢量场。图 3(a) 是该帧原始图像，图 3(b) 是未经过处理的原始运动矢量场，图 3(c) 是经过处理的运动矢量场。从图 3 中可以看出，经过帧内运动估计和运动矢量滤波处理后，消除了大量运动矢量噪声并且保留了更好的运动信息，为下一步深度图推理打下的基础。

(a) 原始图像 (b) 未处理运动矢量场 (c) 处理后运动矢量场

图 3 运动矢量场

3.3 深度图推理和上采样

非关键帧的深度图可以利用 3.1 节和 3.2 节得到的完整并可靠的运动矢量来获得。在 3.2 节中，不同分块的运动矢量都被直接映射到其覆盖的 4×4 块中，然后就可以利用每个 4×4 块的运动矢量推理得到一个缩小为原始图像大小 1/16 的深度图，水平和垂直方向长度分别为原始帧大小的 1/4。该低分辨率深度图像的每个点的深度值都采用运动补偿的方法从前一帧的深度图中获得，如式(5) 所示：

$$D_t^l(p)=\frac{1}{16}\sum_{q\in\Omega}D_{t-1}^h(q) \tag{5}$$

其中：D_t^l 为 t 时刻的低分辨率深度图，D_{t-1}^h 为 $t-1$ 时刻的原始高分辨率深度图，p 和 q 分别为像素位置值。Ω 是像素 q 的范围，该范围可以通过 p 点位置和其对应的运动矢量值进行计算。例如，p 的坐标为 (px,py)，p 点整像素的运动矢量为 (mvx,mvy)，那么 Ω 的范围将是从左上角坐标 $(px\times4+mvx-2,\ py\times4+mvy-2)$ 到右下角坐标为 $(px\times4+mvx+1,\ py\times4+mvy+1)$ 这样一个矩形区域。

然后，为了获得与原始图像大小一样的深度图像，采用一个自适应上采样双边滤波器对式(5) 得到的深度图进行滤波处理，处理算法如式(6) 所示：

$$D_t^h(p)=\frac{\sum_{q\in S}w(p,q)D_t^l(q)}{\sum_{q\in S}w(p,q)} \tag{6}$$

其中：S 是滤波器模板，可以选择不同的模板类型[8]。本文选择半径为 5 的星形模板作为滤波器。$w(p,q)$ 是权重系数，可以由式(7) 计算得到：

$$w(p,q)=\begin{cases}1, & SKIP\ \text{mode}\\ r^{(0,1)}(I^h(p)-I^l(q)), & INTER\ \text{mode}\\ r^{(0,2)}(I^h(p)-I^l(q)), & INTRA\ \text{mode}\end{cases} \tag{7}$$

其中：函数 r 是一个均值为 0，具有不同方差 σ^2 的高斯函数。在滤波过程中，针对不同的宏块编码类型采用不同的滤波强度。如果宏块是帧间编码类型，高斯函数的方差值设为 0.1；如果宏块是帧内编码类型，那么高斯函数的方差值设为 0.2。更高的方差值有助于增加该像素深度值对周围像素深度值的影响。如果宏块类型为 SKIP 模式，不对其进行滤波处理。I^h 是高分辨率图像的亮度值，I^l 是低分辨率图像的亮度值。

每一个 P 帧(非关键帧) 的深度图都利用前一帧的深度图和本帧的运动信息和宏块信息进行推理，最终将得到整个视频序列的深度图。

4 实 验

本文算法采用微软研究院(microsoft research，MSR) 提供的两个公开序列 Ballet 和 Breakdancers 进行测试。这两个序列的分辨率为 1024×768，是专门为测试 3D 视频序列深度图而设计，均包含彩色图像序列和其对应的深度图序列。每个序列包含 100 帧 8 个不同视点的图像。在本实验中，采用第一个视点的彩色图像和其对应的深度图像进行测试。首先，采用目前最流行的视频编码标准 H.264 的参考编码器 JM16.2 对两个序列分别进行编码，序列的每 20 帧图像作为一个帧组(group of pictures，GOP)，每个 GOP 的第一帧作为关键帧并采用 I 帧方式编码，其他帧作为非关键帧并采用 P 帧方式编码。其他

编码参数如下:量化参数(quantization parameter，QP)设为32,运动矢量搜索范围[－32，32],帧率为25帧每秒。本文选择文献[4]中的基于块的运动信息深度推理算法作为对比算法。文献[4]是利用运动信息进行深度图推理的经典算法,可以利用压缩域中的信息。在对压缩域码流进行解码过程中,提取出的运动矢量分别用于本文的深度图推理算法,同时作为文献[4]中所需要的运动信息。

4.1　深度图质量评价

在本实验中,选择峰值信噪比(peak signal to noise ratio,*PSNR*)来作为深度图质量的客观评价标准。*PSNR* 通过均方误差(mean squared error，*MSE*)计算而得到的。而 *MSE* 是通过计算非关键帧的参考基准深度图和推理得到的深度图之间像素的均方差而得到,计算方法如式(8)所示:

$$MSE=\frac{1}{M\times N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|I(x,y)-I'(x,y)|^2 \tag{8}$$

其中:$I(x,y)$ 为参考基准深度图中 $I(x,y)$ 坐标上的深度值,$I'(x,y)$ 是推理深度图中(x,y) 坐标上的深度值,M 和 N 分别是图像的宽度值和高度值。

进而可以得到每帧图像的 *PSNR* 值如式(9)所示,*PSNR* 值越大,深度图的客观质量越好。

$$PSNR=10\times\log_{10}\left(\frac{255^2}{MSE}\right) \tag{9}$$

由于MSR提供的序列 Ballet 和 Breakdancers 本身具有深度图,关键帧深度图采用 Ballet 和 Breakdancers 提供的深度图。非关键帧 Ballet 和 Breakdancers 的深度图作为参考评价基准,与本文算法与文献[4]中算法所得到的推理深度图进行对比。图4画出了本文算法相对文献[4]算法在两个序列20到100帧推理深度图上PSNR值提升情况。可以看到,PSNR最高可以提升达3.7dB,并且平均来说对 Ballet 序列有 1.02dB 的 PSNR 提升,对 Breakdancers 序列有 1.598dB 的 PSNR 提升。

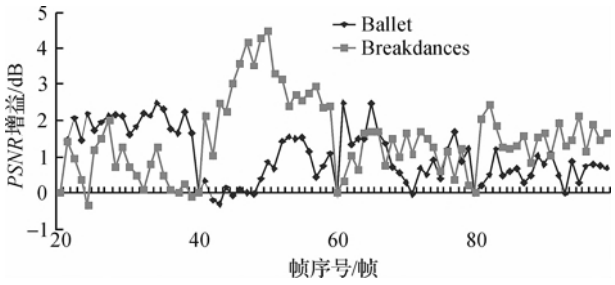

图4　本文算法相对文献[4]算法得到深度图在 *PSNR* 上的增益

4.2　执行时间评价

本文实验所使用的计算机主要配置为 CPU:Intel® CoreTM i5 CPU@2.4GHz,内存:4GB。在此配置和未经过优化的程序执行情况下,本文所使用的算法平均每帧深度图推理所耗时间为 103.6 s,文献[4]平均每帧推理所耗时间为 98s,而大多数基于像素域的深度图推理算法时间都在1000 s/帧以上。图5列出了本文算法相对文献[4]算法在时间上的增长百分比,对于 Ballet 序列和 Breakdancers 序列来讲,分别增加了 2.64%和8.86%,与质量上的提升相比,时间上的增加并不明显。如第3节所描述,本文算法与文献[4]中所增加的时间大部分用于执行帧内宏块的运动估计和运动矢量滤波方面。一般来讲,运动估计过程是更加耗费时间的部分,如图5所示,由于 Breakdancers 序列平均每帧有更多数目的帧内宏块,本文算法的时间增长率也相应有所增加,但增加幅度并不大。

图5　本文算法相对文献[4]算法在时间上的增长百分比

5　结　语

本文提出了一种利用直接从压缩域中获取的信息来进行深度图推理的算法。通过对压缩域中的帧内宏块和帧间宏块分别进行处理,得到了更加完整而准确的运动信息,进而通过运动补偿和上采样双边滤波推理得到深度图。实验结果表明本文方法可以获得图像质量与推理速度的平衡,为 2D 到 3D 视频转换打下良好的基础。

参考文献:

[1] 黄晓军,王梁昊. 2D 视频转换 3D 视频技术概览[J]. 影视制作,2011,17(2):34-36.

[2] VAREKAMP B, BARENBRUG C. Improved depth propagation for 2D to 3D video conversion using key-frames[C]// 4th European Conference on Visual Media Production. London: IET, 2007: 1-7.

[3] WU C, ER G, XIE X, LI T, et al. A novel method for semi-automatic 2D to 3D video conversion[C]// 3DTV Conference: the True Vision - Capture, Transmission

and Display of 3D Video. Istanbul: IEEE. 2008: 65-68.
[4] LIE W N, CHEN C Y, CHEN W C. 2D to 3D video conversion with key-frame depth propagation and trilateral filtering[J]. Electronics Letters, 2011, 47(5): 319-321.
[5] VOSTERS L, DE HAAN G. Efficient and Stable Sparse-to-Dense Conversion for Automatic 2-D to 3-D Conversion [J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2013, 23(3): 373-386.
[6] PHAN R, ANDROUTSOS D. Robust semi-automatic depth map generation in unconstrained images and video sequences for 2d to stereoscopic 3d conversion [J]. Multimedia, IEEE Transactions on, 2014, 16(1): 122-136.
[7] FENG J, CHEN Y, TIAN X. Moving object segmentation algorithm based on cellular neural networks in the H. 264 compressed domain[J]. Optical Engineering, 2009, 48(7): 077001-077001-7.
[8] RIEMENS A K, GANGWAL O P, BARENBRUG B, et al. Multistep joint bilateral depth upsampling [C]// IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2009: 72570M-72570M-12.

Depth Map Inference Algorithm Based on Video Compression Domain

FENG Jie, MA Hanjie

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: In this paper, the depth map inference algorithm in the process of 2D to 3D video conversion was studied, the depth map inference process is based on the unit of macroblock in video compression domain. The research chose the different inference strategies based on different macroblock types. First, motion vectors for the intra-frame macroblocks were calculated by using the neighboring block based motion estimation algorithm. Then, for intra-frame macroblocks, the motion vectors directly extracted are filtered to improve the robustness. Finally, the depth maps are acquired via motion compensation and an up-sample bilateral filter technique. Experimental results demonstrate that the proposed scheme can generate smooth and reliable depth maps with higher quality.

Key words: 2D-to-3D video conversion; depth map; motion vector; compression domain

(责任编辑: 陈和榜)