基于漏斗的实时 VBR 视频最短路径平滑算法①

袁俊杰　徐小良 (杭州电子科技大学 计算机学院　浙江 杭州 310018)

摘　要： 针对实时 VBR 视频流式传输的在线平滑优化问题，提出一种基于漏斗的最短路径平滑算法——SPSF。SPSF 利用滑动窗口对实时 VBR 视频进行分段处理，顺序读取和缓存每帧视频数据至窗口，并基于漏斗原理求解窗口内数据的最短路径。数据填满窗口后根据求得的最短路径进行传输，同时根据路径特征推进窗口滑动进行下一段数据的平滑处理及传输，以此类推完成整个视频平滑传输。实验结果表明，与传统的在线平滑算法相比，SPSF 具有更优的传输比特率峰值、传输比特率谷值、及传输比特率方差；与传统的最短路径算法相比，SPSF 具有更快的最短路径求解速度，提高了视频传输的实时性。

关键词： 实时 VBR 视频；平滑算法；漏斗；最短路径；滑动窗口

Shortest Path Smoothing Algorithm for the Real-Time VBR Video Based on Funnel

YUAN Jun-Jie, XU Xiao-Liang

(Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: To optimize online smoothing of real-time VBR video streaming, this paper proposes a SPSF(Shortest Path Smoothing based on Funnel) algorithm. In the SPSF, the real-time VBR video is piecewise processed by sliding window. Each video frame is read and cached into the window and the shortest path for the data transmission in the window is calculated based on funnel principle. When the window is filled with data, data start transmission according to the obtained shortest path. Window simultaneously slides according to the characteristics of the path to start smoothing and transmission of the next piece, and so complete the smooth transmission of the entire video. The experimental results show that compared with conventional online smoothing algorithm, SPSF has better bit-rate peak, valley, and variance. Compared with traditional shortest path algorithm, SPSF takes less time in computation of shortest path so as to be able to meet the real-time requirement.

Keywords: real-time VBR video; smoothing algorithm; funnel; the shortest path; sliding window

1　引言

由于 VBR(可变比特率)视频具有较强的突发性比特率变化特性，使有限带宽、受限缓冲的网络传输技术复杂化。为了有效利用传输带宽资源和提高视频服务质量，VBR 视频在传输前必须进行平滑处理。

VBR 视频根据视频源的性质分为预存储视频和实时视频。针对预存储 VBR 视频的平滑算法称为离线平滑算法[1]，这类算法在事先知道视频完整信息的前提下，将视频的整个传输过程分为几个具有恒定传输比特率的段。通过各自不同的分段方法，这类算法能够获得最小的比特率峰值[2,3]，最小的比特率变化次数[4]或最小的比特率方差[5]。目前，较理想的一种离线平滑算法是 SPS(Shortest Path Smoothing)算法[6]，它将平滑问题转换为二维空间最短路径问题，通过求解最短路径来获得最小比特率峰值、最大比特率谷值、及最小比特率方差。但是 SPS 算法的时间复杂度比较高，也不适用于实时 VBR 视频的在线平滑。针对实时 VBR 视频的平滑算法称为在线平滑算法[7]，与离线平

① 基金项目:浙江省科技计划重点科研项目(2008C21082);浙江省重大科技专项(2007C11070)
收稿时间:2009-11-02;收到修改稿时间:2009-11-23

滑不同，在线平滑不知道完整的视频信息，并且对延时有严格的限制。因此，在线平滑算法采用滑动窗口方法分段缓存实时 VBR 视频，在各个分段中再采用 SPS 算法进行离线平滑处理[8]。但是这种方法没有考虑窗口之间最短路径的关系，不能同时兼顾平滑质量与传输实时性。

本文提出一种基于漏斗的最短路径平滑算法(SPSF — Shortest Path Smoothing based on Funnel)，算法利用滑动窗口方法缓存实时产生的视频帧数据，同时基于漏斗原理快速计算窗口内累积数据的最短路径，当缓存数据填满窗口时按照窗口内的最短路径传输视频，并根据路径特征向后滑动窗口。新算法有机结合了滑动窗口与最短路径平滑算法，将有效提高实时 VBR 视频的平滑质量及传输实时性。

2 平滑模型

设一个实时 VBR 视频由 n 帧数据组成，fi 为第 i 帧数据的字节数，T 为帧单位时间。为了保证视频的持续播放，服务器累积发送的数据量 $S(t)$必须满足下面约束公式：

$$U(t) \le S(t) \le O(t) \tag{1}$$

式(1)中 $O(t)$、$U(t)$为传输上下限，若 $S(t)$低于下限 $U(t)$，说明数据不能及时发送至客户端，播放将出现停顿。若 $S(t)$超过上限 $O(t)$，说明发送的数据量太大导致客户端缓冲区溢出，播放数据丢失。$U(t)$、$O(t)$可由公式2、3计算得出：

$$U(t)=\sum_{i=1}^{k} f_i \qquad k=\lfloor t/T \rfloor \tag{2}$$

$$O(t)=U(t)+b \tag{3}$$

$U(t)$和 $O(t)$呈阶梯形，两者闭合构成一个单调多边形，而 $S(t)$为此多边形内以 s 为起点 e 为终点的一条路径(图 1)。

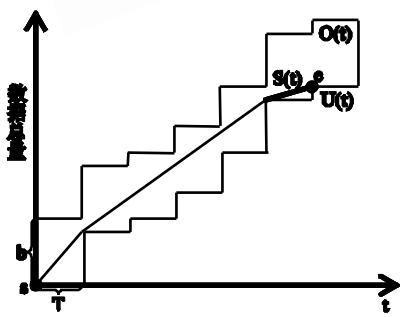

图 1 平滑模型

假设一任意可行的传输方案 S 由 M 段直线构成，M 段直线的斜率(传输比特率)分别为 $\{y_1, y_2, y_3, \cdots, y_m\}$。若存在 S*为最优传输方案，由 K 段直线构成，K 段直线斜率为$\{x_1, x_2, x_3, \cdots, x_k\}$。那么 S*一定拥有比 S 更小的传输比特率峰值、更大的传输比特率谷值及更小的传输比特率方差，可以用公式表示为：

$$max(x_i) \le max(y_i); min(x_i) \ge min(y_i) \tag{4}$$

$$\sum_{i=1}^{K}(x_i-\bar{x})^2/K \le \sum_{i=1}^{M}(y_i-\bar{y})^2/M \tag{5}$$

其中 $\bar{x}=\sum x_i, \bar{y}=\sum y_i$ 。

而在 $U(t)$和 $O(t)$构成的单调多边形空间中，从起点 s 至终点 e 的最短路径正好能够满足公式(4)、(5)。因此，求最优平滑传输方案的问题本质上即为求最短路径的问题[6]。

3 SPSF算法

本文提出的 SPSF 在线平滑算法，首先根据滑动窗口对实时 VBR 视频进行分段处理，然后基于漏斗[9]原理计算窗口内累积数据的最短路径，算法的具体描述如下。

3.1 算法步骤

SPSF 算法的步骤如下：

①顺序读取一帧视频数据并缓存于窗口中。若没有数据可读，跳到第⑥步。

②基于漏斗原理计算窗口中累积数据的最短路径。

③若窗口数据填满了进入第④步，否则返回第①步。

④按照求得的最短路径传输窗口内的视频数据

⑤根据路径特征向后滑动窗口，返回第①步。

⑥按照求得的最短路径传输窗口内的数据，程序结束。

3.2 基于漏斗求解最短路径

如图 2 所示，$O(t)$和 $U(t)$构成的区域可以看作是一个单调多边形。我们用$\Pi(s,w)$来表示多边形内点 s 至点 w 的最短路径，那么$\Pi(s,e)$即为所求最短路径。显然，$\Pi(s,e)$只可能与多边形的凹点(例如 v)相交，不可能与凸点(例如 z)相交。因此，计算最短路径只需要考虑凹点。假设已知 $\Pi(u,s)=[u,s]$及 $\Pi(s,w)=[s,v,w]$，那么漏斗就是由$\Pi(u,s)$以及$\Pi(s,w)$构成的 V

形结构，用 F_{uw}表示，而 s(V 形底端)为漏斗的底。F_{uw}是由最初漏斗经过数次更新后生成的，最初的漏斗由起始点 s 与 $O(t)$、$U(t)$的第一个凹点的连线构成的，如图中虚实线所示。

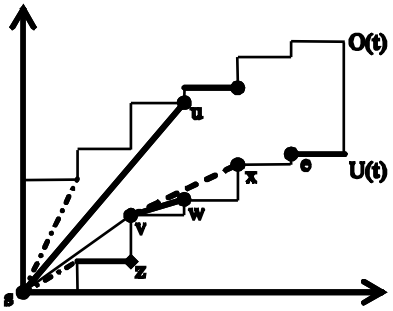

图 2　漏斗

在现有漏斗 F_{uw}的基础上，计算出漏斗底至下一凹点 x 的最短路径$\Pi(s,x)$，那么漏斗就更新为 $F_{ux}=\Pi(u,s)+\Pi(s,x)$，依次类推计算后续凹点直至终点 e，即可得到 F_{*e}，而 F_{*e}的下半部分即为所求的$\Pi(s,e)$。

在 F_{uw}的基础上，计算$\Pi(s,x)$的方法如下。首先，如果 s 到 x 的直线不与 F_{uw}相交，那么$\Pi(s,x)$就是直线 sx。否则，在 Fuw 中寻找一个点 v_i，使得 v_{ix}与 F_{uw}相切。用公式可以表示为：

$$tg(v_{i-1}v_i)\geq tg(v_ix)\geq tg(v_iv_{i+1}) \quad (6)$$

其中 tg 表示线段的斜率，$v_{i-1}v_i$以及 v_iv_{i+1}为 F_{uw}中 v_i的前后线段。找到点v_i后连接v_{ix}，那么 $\Pi(s,x)=[s,\cdots vi,x]$。

在最短路径求解过程中还需要考虑漏斗底的更新。漏斗的更新有两种，第一种如之前所讲，只对漏斗的上下两条路径进行删除添加操作，而第二种需要更新漏斗的底。如图 3 所示，在漏斗 F_{ux}的基础上计算$\Pi(y,s)$。通过查找得到满足公式 6 的点 v，因此$\Pi(y,s)=[y,v,s]$。由于$\Pi(y,s)$与$\Pi(s,x)$相交于点 v，那么新漏斗 F_{yx}的底需要更新为 v，同时线段$[s,v]$作为最短路径的一部分被保存起来。

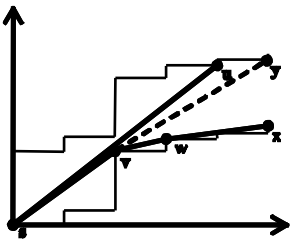

图 3　漏斗底更新

3.3 滑动窗口

使用漏斗方法进行在线平滑计算时还需要结合滑动窗口方法。与离线平滑不同，在线平滑需要在获取视频帧数据的同时计算最短路径，因此服务器需要缓存一定时间单位的帧数据，计算出这段数据的最短路径后再开始传输，而这段帧数据就是窗口。如图 4 中所示，数据传输进行至 t 时刻，同时已获取 t 后 W 单位时间的帧数据。由于获取帧数据的同时计算最短路径，此时已经得到窗口内的最短路径，继而可以按此路径开始传输。如果在窗口中漏斗底没有更新，那么窗口向后滑动 W 变为 $t+W$ 至 $t+2W$，在传输的同时计算新窗口中的最短路径，但是新旧窗口的最短路径并不连续，即$\Pi(t,t+W)+\Pi(t+W,t+2W)\neq\Pi(t,t+2W)$。相反，如果发现在 $t+\alpha$ 时刻发生漏斗底更新，窗口向后滑动 α，并将延续旧窗口的结果计算新窗口的最短路径，那么$\Pi(t,t+\alpha)+\Pi(t+\alpha,t+\alpha+W)=\Pi(t,t+\alpha+W)$，最短路径是连续的。

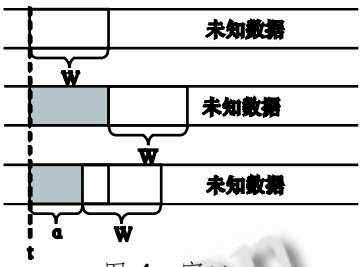

图 4　窗口

3.4 算法分析

SPSF 算法中漏斗的更新操作都是在两端进行的，因此可以将漏斗存储在一个双端队列中。例如漏斗 F_{uw}存储为$[u_j,u_{j-1},\cdots,u_1,c,w_1,\cdots,w_{k-1},w_k]$，其中 c 为 F_{uw}的底，$\Pi(u,c)=[u_j,u_{j-1},\cdots,u_1,c]$，$\Pi(c,w)=[c,w1,\cdots,w_{k-1},w_k]$，$(u_j=u,w_k=w)$。显然，漏斗中线段的斜率是递减的，因此寻找切点可以使用二分查找。为了方便二分查找以及添加删除操作，用循环数组来实现双端队列。若漏斗长度为 M，那么寻找切点的时间复杂度为 $O(\log M)$。如果一个视频的总帧数为 N，那么 SPSF 算法的时间复杂度就 $O(N\log M)$。最坏情况下漏斗的长度从 1 到 $2N$，算法的复杂度就是$\sum\log i=O(\log(N!))<O(N\log N)$。

SPSF 算法需要合理地设置窗口大小。窗口太小，平滑效果将不够理想，窗口太大会造成较大延时。理想情况下，每一个窗口内都存在漏斗底更新，那么各

个窗口的最短路径就是连续的，在线平滑就可以达到离线平滑的效果。同时，SPSF 不同于传统的在线平滑算法，传统算法中各窗口独立进行平滑计算，因而窗口重叠部分需要重复计算最短路径,而 SPSF 中窗口间可以连续计算最短路径，在保证平滑质量的情况下提高了平滑处理性能。

4 实验

本文中，我们采用一个 MPEG2 编码的视频文件作为样本 VBR 视频，视频时长 40 分 10 秒，总帧数为 64747，帧率为 25。该视频中最大帧的数据量为 82KB，实验中设置的客户端缓冲尺寸为 100KB。首先，对样本视频进行 SPSF 离线平滑处理(即窗口大小设为视频总长)，与传统 SPS 算法及原始传输进行比较实验。然后，进行 SPSF 在线平滑平滑处理，与传统在线平滑算法(窗口滑动距离设为窗口尺寸的 1/2)进行比较实验。

图 5 显示了未经平滑和经 SPSF 离线平滑后的视频传输率情况，图中细的曲线为未经平滑的视频传输率，粗的曲线为 SPSF 离线平滑后的传输率。从图中看出，原始传输率分布范围为 183～1118KB，SPSF 平滑后分布范围为 432～972KB，范围缩减约 50%，效果明显。

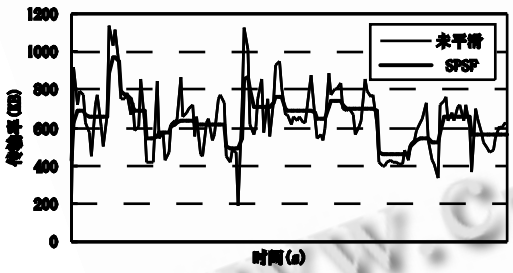

图 5 离线平滑

SPSF 离线平滑的时间复杂度为 O(N log M)，其中 M 为漏斗的长度。图 6 显示的是漏斗最大长度与客户端缓冲的关系。可以看出，漏斗最大长度随着客户端缓冲区尺寸的变大而增长。但是，当缓冲区尺寸超过一定程度，它将保持在 25。对大量不同 VBR 视频进行实验后发现，漏斗最大长度一般在 20 左右，并且与视频总帧数 N 无关，可以看作是常量。因此，SPSF 可以认为是一种线性时间算法，与时间复杂度为 O(KN)的传统 SPS 算法相比有明显优势。

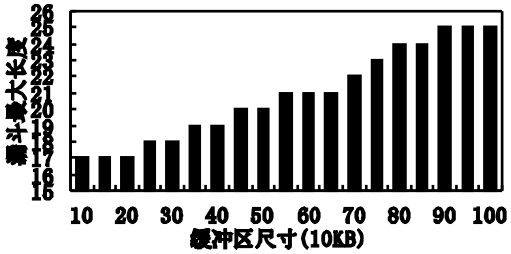

图 6 漏斗最大长度与缓冲关系

图 7 是经 SPSF 在线平滑后的实验结果，图中细的曲线是窗口大小设为 2 秒(50 帧)的在线平滑结果，粗的曲线是窗口大小设为 4 秒(100 帧)的在线平滑结果。显然，窗口设置越大，平滑效果越好，当窗口为 4 秒时，传输率分布为 410～972KB，其效果已接近于离线平滑的效果。

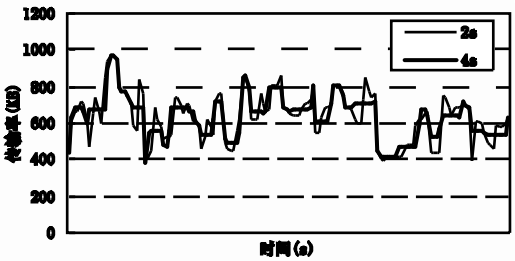

图 7 在线平滑

图 8 与图 9 分别显示了 PAR(Peak Average Rate)和 COV(Coefficient Of Variation)与窗口尺寸的关系，PAR 反映了传输比特率峰值的情况，COV 反映了传输比特率方差的情况。图中粗线为 SPSF 在线平滑的结果，细线为传统在线平滑的结果。从图中可以看出，SPSF 的平滑效果优于传统在线平滑算法，而且在窗口变大的过程中，PAR 值与 COV 值有减小的趋势。但当窗口增加到一定程度时，曲线不再发生变化，说明此时在线平滑已经达到离线平滑的水平。当然如果要让这两个值继续下降则需要进一步加大客户端缓存。

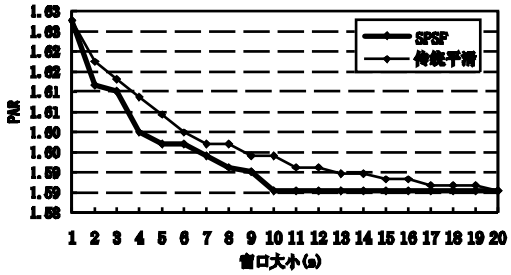

图 8 PAR 和窗口尺寸的关系

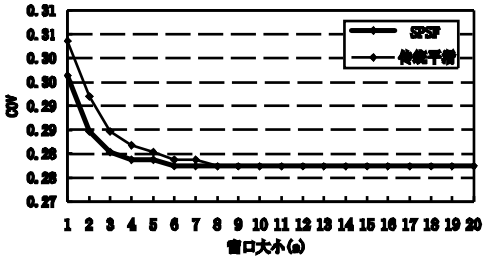

图 9　COV 与窗口尺寸关系

5　结论

本文提出了一种基于漏斗的实时 VBR 视频最短路径平滑算法。算法利用滑动窗口方法分段处理实时 VBR 视频，基于漏斗原理计算窗口内视频数据的最短路径，滑动窗口时考虑先前的路径特征以保证完整视频的最优平滑。新算法能有效降低实时 VBR 视频传输比特率峰值，提高传输比特率谷值，减小传输比特率方差。在合理设置窗口大小的情况下，新算法的在线平滑效果能够接近于离线平滑效果。与传统 SPS 算法相比，新算法的时间复杂度较低，能够有效减轻流媒体服务器负担，有利于提高视频传输的实时性。

参考文献

1 Feng W, Rexford J. A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecoded Compressed Video. Proc. of INFOCOM 97, Sixteenth Annual Joint Conference of IEEE Computer and Communicaions Societies. 1997.58－66.

2 Feng W, Sechrest S. Smoothing and buffering for delivery of prerecorded compressed video. Proc. of the IS&T/SPIE Symposium on Multimedia Computing and Networking. San Jose, CA, 1995.234－242.

3 Feng W, Sechrest S. Critical bandwidth allocation for delivery of compressed video. Computer Communications, 1995,18(10):709－717.

4 Feng W, Jahanian F, Sechrest S. Optimal buffering for the delivery of compressed prerecorded video. ACM/Springer-Verlag Multimedia Systems Journal, 1997,8(9):297－309.

5 Salehi JD, Zhang ZL, Kurose J, Towsley D. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. IEEE/ACM Transactions on Network, 1998, 6(4):379－410.

6 谢建国,姜灵敏,陈松乔. VBR流式视频的最短路径率平滑传输算法.计算机学报, 2004,27(3):357－364.

7 Rexford J, Sen S, Dey J, et al. Online smoothing of live,Variable-bit-rate video. Proc. of the 7th Workshop on Network and Operating Systems Support for Digital Audio and Video. St Louis, MO, 1997.249－257.

8 Subhabrata S, Jennifer LR, Jayanta KD, et al. Online Smoothing of Variable-Bit-Rate Streaming Video. IEEE Transactions on Multimedia, 2000,2(1):37－47.

9 Guibas L, Hershberger J, Leven D, Sharir M, Tarjan R. Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica, 1987,2:209－233.