

文章编号:1001-9081(2006)08-1863-03

具有可伸缩性的增强型高级锥形广播算法

王 勇, 吕 钊

(华东师范大学 计算机科学系, 上海 200062)

(ywang@ica.stc.sh.cn)

摘 要: 镜像金字塔广播算法 (MPB) 采用镜像的方法来减少金字塔算法 (PB) 对用户缓存的要求。EMPB 算法和 APB 算法则对该算法做了进一步改进以提高其性能。但 APB 算法并不具备可伸缩性和最好的性能。通过对这些算法的分析, 提出了高效可伸缩高级锥形算法 SAPB。该算法能适应不同用户缓存限制, 并通过可伸缩参数的选择可获取比上述算法更好的性能, 使 VoD 系统具有很好的灵活性和较强的用户环境适应能力。

关键词: 视频点播; 锥形广播; 镜像锥形广播; 高级锥形广播

中图分类号: TP37; TP393.09 **文献标识码:** A

Enhanced advanced pyramid broadcasting with scalable capability

WANG Yong, Lü Zhao

(Department of Computer Science, East China Normal University, Shanghai 20062, China)

Abstract: Mirrored-Pyramid Broadcasting (MPB) can reduce the client buffer requirement of Pyramid Broadcasting (PB), and Enhanced Mirrored-Pyramid Broadcasting (EMPB) and Advanced Pyramid Broadcasting (APB) are the improved ones. But APB does not have the flexibility or best performance. Based on the analysis of these schemes, a new broadcasting scheme of SAPB was proposed. It can meet the various client buffer requirements, and use a scalable parameter to get the scalable capability and better performance. By adopting this scheme, VoD system will be more flexible and adaptive.

Key words: Video on Demand (VoD); pyramid broadcasting; mirrored-pyramid broadcasting; advanced pyramid broadcasting

0 引言

相对于广播电视系统, VoD 系统因提供用户可以在任何时候都收看他们喜欢的视频节目的能力, 而得到广泛的应用。特别是最近广为流行和大规模开展的 IPTV 服务。但是这种服务的实现对服务器的性能和网络都有非常高的要求。典型的情况下, 每个用户访问的视频都要占用独立的网络带宽和服务器资源。大量的研究发现, 在 VoD 系统中, 40% ~ 60% 的用户点播都是对少量视频节目 (10% ~ 20%) 的访问^[1]。因此研究人员提出了合并用户的点播需求的视频广播算法来实现流行视频节目的发布。这类广播算法将这些视频节目按照固定的方法在信道中周期性地广播, 当用户访问视频的时候, 直接从最近的视频节目起点处接收和播放该视频。这种方法在理论上可以用少量的信道满足无限多个用户访问同一视频节目的需求。大量被开发出来的广播算法通常都着眼于如何提高网络带宽的利用率^[2], 但网络带宽利用率的提高将对用户端资源提出更多的要求, 其中包括对用户缓冲区的要求。

1 锥形广播算法和镜像广播算法介绍

1.1 锥形广播算法 (PB)

PB 算法是第一个被提出的用于减少用户等待时间和网络带宽的广播算法^[2]。其基本思想是:

1) 在服务器端: 服务器分配网络带宽 B 用于广播某个视频, 并将这些带宽分成 N 个信道。每个信道带宽为 B/N (该值大于 1); 将长度为 L 的该视频也分为 N 个尺寸递增的视频段

(每个段用 $L_i (1 \leq i \leq N)$ 表示), 视频分段满足如下关系: $L_i = \alpha L_{i+1} (\alpha > 1)$, $L = L_1 + L_2 + \dots + L_N$; 每个视频段 L_i 放在信道 i 中循环广播。

2) 在用户端: 当用户接入广播信道后, 从最近的 L_1 视频段的起始处开始下载并播放该视频段, 同时也在其他逻辑信道开始下载别的视频段。

在该算法中, α 值的选择是一个关键。因为用户的最大等待时间是第一个视频段被完整下载的时间, 当使用的 α 值越大, L_1 长度越小, 则用户的等待时间越小。但过大的 α 值将导致用户不能够连续播放该视频。连续播放条件为: $B/N > 1$ 时, 用户最迟必须在 L_i 视频段播放完成的时候能够开始下载第 L_{i+1} 段视频。而且用户在播放视频的时候, 将同时在多条信道下载数据, 因此还没有来得及播放的数据必须由用户端缓存。该算法对于用户端缓存需求非常高, 最大缓存需求达到视频长度的 70%。

1.2 镜像锥形算法 (MPB)

文献[3]根据 PB 算法产生对用户缓存大量需求的原因, 提出了镜像分配视频段大小的方法来减小对用户端缓存的需求。基本思路和 PB 算法一致, 都是将服务器带宽分成多个等宽的信道, 然后将视频也分成多个段。但是其使用的信道大小固定, 且视频分段方法与 PB 算法不同。该算法信道的分配和视频分段的方法如下: 设视频的播放速率 (也称为消耗带宽) 为 b ; 服务器带宽分成 N 个带宽 $2b$ 的信道; 视频也分成 N 个段, 在前面一半信道内广播的视频段长度逐道倍增, 后面一半信道内广播的视频段逐道递减。分段公式如下 (为了

收稿日期: 2006-02-20; 修订日期: 2006-04-09

作者简介: 王勇 (1974-), 男, 重庆人, 工程师, 博士研究生, 主要研究方向: 流媒体编码与传输、数字电视; 吕钊 (1970-), 女, 四川江油人, 副教授, 博士, 主要研究方向: 多媒体技术、CSCW、信息安全。

说明方便,这里只分析 N 为奇数的情况):

$$L_i = \begin{cases} L_1 & \text{if } i = 1 \\ 2L_{i-1} & \text{if } 2 \leq i \leq q+1 \\ L_{i-1}/2 & \text{if } q+2 \leq i \leq N \end{cases} \quad (1)$$

其中 $q = (N-1)/2$ 。当 N 为 5 时候,每个信道内广播的视频段长度为 $L_1, 2L_1, 4L_1, 2L_1, 1L_1$ 。

缓存需求分析:如图 1,每个视频下载信道带宽为 $2b$,而视频的播放(消耗)速度为 b ,将信道分为多个时隙组成,每个时隙为 $L_1/2b$ 时间长度。从图 1 可以看出,任何一个数据段 L_{i+1} ($2 \leq i+1 \leq q+1$) 的下载时间等于其前一段 L_i 的播放时间:即 $L_{i+1}/2b = 2L_i/2b = L_i/b$ 。在任何数据段 L_i 段播放期间都会(且也只能)在信道 $i+1$ 中遇到一个 L_{i+1} 的起点,如图 1 中 L_2 的播放期间只会遇到阴影区域的 L_3 的起点。因此,必须在此 L_{i+1} 数据段的起点处进行 L_{i+1} 数据段下载,才能保证 L_i 段播放完成后能连续播放 L_{i+1} 数据段,且对用户缓存需求最小(越迟下载后续播放的数据段,对用户缓存要求越小)。当 $2 \leq i+1 \leq q+1$,数据段 L_{i+1} 最早将在数据段 L_{i+1} 被播放了 $L_i/2$ 长度的时候被下载完。如图 1 所示:阴影区域的 L_3 处于最早被下载完的位置;如果整个视频播放时间提前 1 个时隙, L_3 将在被播放了 L_1 长度之后下载完;如果整个视频播放时间延迟一个时隙, L_3 可在阴影区域后面的时隙中被下载;因此该位置是最早被下载完的位置。此时用户缓存需求为最大值,由于用户下载的总数据量为 $L_1 + \dots + L_{i+1}$,消耗掉的数据段为 $L_1 + \dots + L_i + L_i/2$,所以缓存最大需求为两者之差: $L_{i+1} - L_i/2$ 。当 $i = q$ 的时候,用户缓存需求最大为 $L_{q+1} - L_1/2$ 。而从 $q+2$ 开始的信道中,数据段长度依次减半;则信道 j ($j > q$) 的数据段 L_j 的播放期间将在 $j+1$ 信道遇到视频段 L_{j+1} 的 4 个起点,可以在最后一个起点处开始下载 L_{j+1} ;而此时已经消耗掉 $3L_{j+1}/2$ 的缓冲数据,所以不会引起缓冲数据量的增加。因此该算法最大的缓冲需求为 $L_{q+1} - L_1/2$ (最大数据段被下载完的时候)。

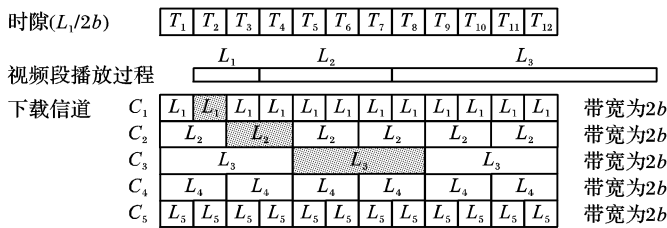


图 1 MPB 算法广播 5 个视频段的下载和播放过程

该算法通过减小后面段的长度来减小对用户端缓冲的需求,但是这种方法对服务器带宽浪费非常严重,一共需要消耗 $N * 2b$ 的服务器带宽才能使得用户的最大访问等待时间为 $L_1/(2b)$ 。

1.3 增强镜像锥形算法 (EMPB)

针对上述算法的带宽浪费问题,文献[4]对 EMPB 算法进行了改进。主要改进内容如图 2 所示为:前面 $q+1$ 信道和原算法一致,而将 L_{q+2} 到 L_p 段合并成 1 段,并放入到带宽为 $2b$ 的第 $q+2$ 信道中;原来的 $q+3$ 信道及以后的信道就不再需要。该算法使用的服务器网络带宽从原来的 $N * 2b$ 减小到 $((p-1)/2 + 1) * 2b$,而用户的缓冲需求和访问等待时间并不会发生变化。在该算法中存在 $L_{q+2} + L_{q+3} + \dots + L_N = L_{q+1} - L_1$,表明在第 $q+2$ 信道中仍然还存在空的时隙,对网络带宽仍然有一定的浪费。该算法描述为:使用 N 个带宽为 $2b$ 的服务器信道时,视频分为 N 段,前面 $N-1$ 段视频倍增,最后一个视频段长度为 $L_{N+1} - L_1$ 。因此整个视频长度 $L = (L_1 + \dots + L_{N-1}) + (L_1 + \dots + L_{N-2}) = (3 * 2^{N-2} - 2)L_1$,用户等待时间与整个视频

时间比用 W 表示为: $W = (L_1/2b)/(L/b) = 1/(6 * 2^{N-2} - 4)$ 。

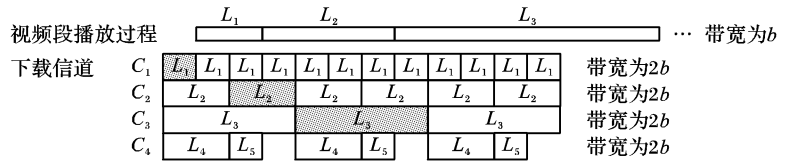


图 2 EMPB 算法广播 5 个视频段的下载和播放过程

2 高级镜像锥形算法和高级锥形算法

文献[5]分析了 EMPB 对客户缓冲需求产生的过程:当用户以 $2b$ 的速度下载视频的同时以 b 的速率消耗视频时,最大段尺寸 L_{q+1} 给用户的缓冲造成最大的消耗。为了解决这个问题而提出了高级镜像锥形算法 (AMPB),主要的思路是:将最后的 5 个信道带宽减小为 b ,用来广播最后的 3 个视频段;使得最后 5 个信道中的视频段被下载的时候不会增加对用户缓存的消耗。但是该算法在做用户播放连续性分析中存在问题,因此结论是不正确的。文献[6]提出的高级锥形算法 (APB)对该问题进行了修正,并解决了 EMPB 算法对最后一个信道中的空闲时隙对带宽的浪费,获得了预期的性能。APB 算法如下:服务器端带宽分为 N 个信道,其中最后 5 个信道的带宽为 b ,其余信道带宽为 $2b$;视频一共分为 $N-2$ 段;第 1 到第 $N-5$ 段倍增;倒数第 3 段与倒数第 4 段大小相同;最后 2 段大小相同且为倒数第 3 段的 2 倍。分段公式描述为:

$$L_i = \begin{cases} L_1 & \text{if } i = 1 \\ 2L_{i-1} & \text{if } 2 \leq i \leq N-5 \\ L_{N-5} & \text{if } i = N-4 \\ 2L_{N-5} & \text{if } i = N-3 \\ 2L_{N-5} & \text{if } i = N-2 \end{cases} \quad (2)$$

将第 1 到第 $N-4$ 视频段分别放入前 $N-4$ 个信道内循环广播;最后 2 个视频段都分别在 2 个带宽为 b 的信道循环广播,同一个视频段在 2 个信道中广播时间偏移 L_{N-5}/b 时间,如图 3 所示。

根据该算法有:

$$\begin{aligned} L &= L_1 + L_2 + \dots + L_i + \dots + L_{N-2} \\ &= (2^{N-5} - 1 + 2^{N-6} + 2^{N-5} + 2^{N-5}) * L_1 \\ &= (7 * 2^{N-6} - 1) * L_1; L_{N-5} \\ &= 2^{N-6} * L_1 \end{aligned}$$

而消耗的总网络带宽为 $n * b$ 的时候有 $n = 2(N-5) + 5$,所以用户等待时间与视频总时间之比为:

$$\begin{aligned} W &= (L_1/2b)/(L/b) \\ &= 1/2 * (7 * 2^{N-6} - 1) \\ &= 1/2 * (7 * 2^{(n-7)/2} - 1) \end{aligned}$$

因为前面的 $N-5$ 个信道和 EMPB 算法一致,根据 EMPB 算法对缓冲需求的分析可以得到,本算法对用户缓存的需求为: $L_{N-5} - L_1/2$ 。缓存需求与视频长度的比例为 $Buff = (L_{N-5} - L_1/2)/(7 * 2^{N-6} - 1)L_1 = (2^{N-6} - 1/2)/(7 * 2^{N-6} - 1)$,当 N 趋向于无穷大的时候, $Buff$ 接近 $1/7$ 。

3 对 APB 算法的分析和提出新的算法 SAPB

图 3 可以等价变换成图 4。在图 4 中可以看出,变换过程将原来图 3 中的最后 2 个视频段 L_3 和 L_4 分成 4 个相等的视频段 L_3, L_4, L_5 和 L_6 。变换后,从 L_2 开始的所有视频段的结束处,都能在下一个信道找到下一个视频段的开始,并且这些信道的带宽和视频的消耗带宽相等。因此,对这些视频段的下载不需要额外的缓存。变换后的视频分段方法和广播方法

的性能与 APB 算法完全相同。

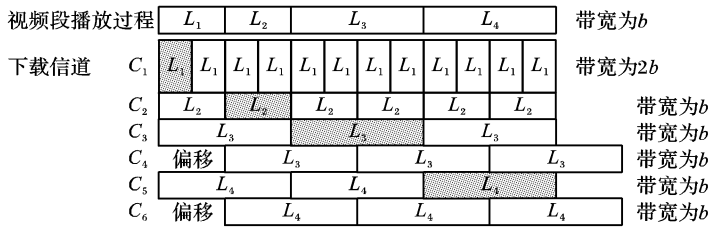


图3 APB算法广播5个视频段的下载和播放过程

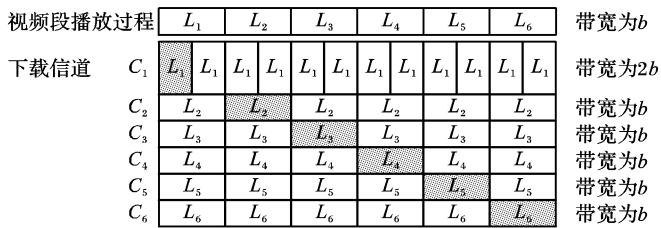


图4 等价变换的APB算法广播5个视频段的下载和播放过程

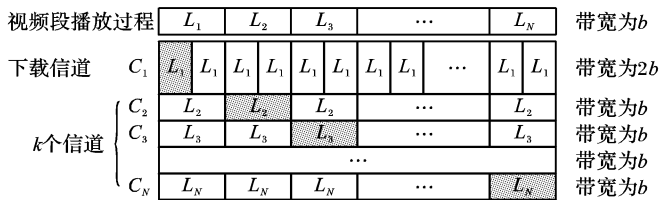


图5 SAPB算法广播N个视频段($k = N - 1$)的下载和播放过程

从上面的分析,可以得出一个规律:在MPB算法中的 $q + 1$ 信道后增加 k 条带宽为 b 的信道,并在其中放入与 L_{q+1} 视频段相等大小的视频段,能够减少对用户缓冲的需求;并且 k 值越大,用户缓存需求越小。但 k 值越大需要的信道数也会越多,而且带宽利用率降低。因此,在实际使用中,要根据用户的缓存能力和系统中实际可用的信道数来决定该值。为了获得尽量好的性能, k 值通常的选取方法为:在满足可用信道数和用户缓存限制的情况下取最小的值, k 值可灵活选取使得本算法具有满足各种应用环境需求的可伸缩能力。

根据这个规律,本文提出可伸缩性锥形算法SAPB,算法内容如下:

1) 服务器共使用 N 个信道,其中后面 k 个信道的带宽为 b ,前面 $N - k$ 个信道的带宽为 $2b$;

2) 视频分段公式为:

$$L_i = \begin{cases} L_1 & \text{if } i = 1 \\ 2L_{i-1} & \text{if } 2 \leq i \leq N - k \\ L_{N-k} & \text{if } i > N - k \text{ and } i \leq N \end{cases} \quad (3)$$

将 L_i 视频段在信道 i 中循环广播。

3) 在用户端,按照EMPB的算法接收前面 $N - k$ 个信道的视频段;后面的 k 个信道的视频段在上一个视频段播放完成后再接收。

根据上面的算法有:

$$\begin{aligned} L &= L_1 + L_2 + \dots + L_{N-k} + \dots + L_N \\ &= (2^{N-k-1} + k * 2^{N-k-1}) * L_1 \\ &= ((2 + k) * 2^{N-k-1} - 1) * L_1 \end{aligned}$$

$$L_{N-k} = 2^{N-k-1} * L_1$$

因此APB算法等价于本算法在 $k = 5$ 时的特例。而用户等待时间与视频总时间的比 $W = (L_1/2b)/(L/b) = 1/((2 + k) * 2^{N-k-2})$ 。用户缓存的需求根据EMPB算法中的用户缓冲需求分析为 $2^{N-k-1} * L_1 - L_1/2$,则用户缓冲需求与视频长度的比为: $Buff = (2^{N-k-1} * L_1 - L_1/2)/L = (2^{N-k-1} - 1/2)/((2 + k) * 2^{N-k-1} - 1)$ 。

4 性能仿真

在本节中对PB、EMPB和SAPB在用户磁盘需求和用户访问等待时间方面的性能进行了比较。由于APB算法与SAPB算法在参数 $k = 5$ 时的性能相同,PB算法在 $\alpha = 2$ 时的性能与SAPB算法取 $k = 2$ 时相同,本文分别计算了 $k = 1, k = 3, k = 5, k = 7$ 和 $k = 2$ 时的性能。在图6中,横坐标为服务器分配的带宽,以视频消耗率 b 为单位;纵坐标为用户等待时间与视频总时间之比。由于PB算法对用户缓存需求超过视频总长度的70%,而远远大于MPB和EMPB算法。图7只列出了EMPB算法与SAPB算法取不同参数时对用户缓存需求的比较。在图7中,横坐标与图6一致,纵坐标为用户缓存需求与视频总长度之比。从图中可以看出,当 k 值越大,服务器带宽利用率越低,而用户缓存需求越小。相对于EMPB算法,SAPB算法在相同的服务器带宽的情况下,当 $k < 5$ 时,本算法不但比EMPB算法对用户缓存需求小,而且用户等待时间也有一定程度的减小;当 $k \geq 5$ 时,对用户缓存需求的更多减小,但用户等待时间增加比较明显。因此在应用此算法的时候,需要根据实际情况选择最小的 k 值,以实现在满足各种不同资源限制的情况下获得最好的性能。

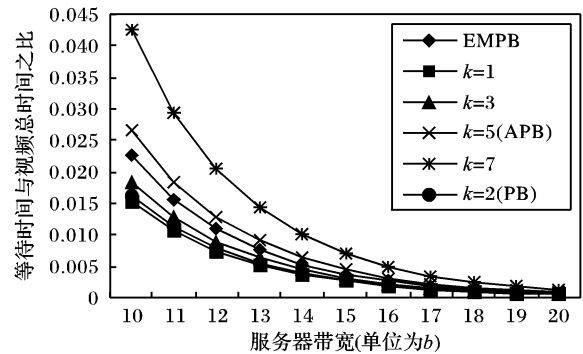


图6 用户等待时间比较

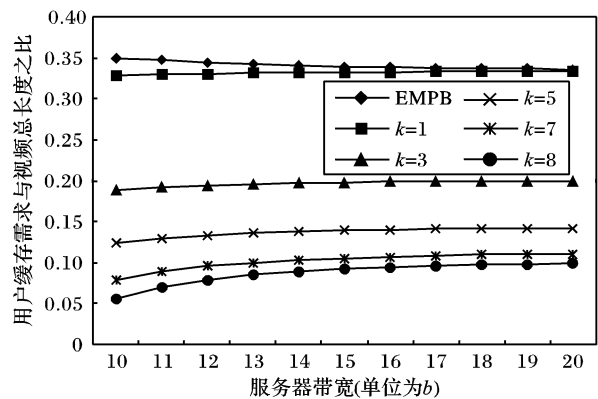


图7 用户缓存需求比较

5 结语

本文分析了AMPB和APB算法对镜像广播算法进行改进的原理,提出了对EMPB在减少用户缓存需求方面的可伸缩算法SAPB。并分析和比较了这几种算法的性能,结果显示用户缓存能力的限制将带来网络带宽利用率的降低,但采用合理的参数选择和算法设计,可以尽可能地利用用户缓存能力来提高网络利用率。本文提出的算法能适应不同的用户缓存能力,通过调整可伸缩参数 k 来获取最好的性能,为VoD

(下转第1869页)

部窗口内按照 1.1 节第二种方法计算了均匀度映像(图 3(c))。图 3 最后一行是在 23×23 的局部窗口内按第二种均匀度定义方法计算的图像色彩均匀度映像。这些结果表明: 1) 色彩均匀度确实可以用来描述纹理特征。2) 但相对之下, 用质心与窗口几何中心的距离来度量均匀度(第二种方法)更好, 特别是对于局部纹理特征分析, 利用色彩密度的方差来衡量均匀度(第一种方法)所取得的效果很不理想。原因是, 在过小的子区域上计算色彩密度从统计来看并不可靠, 所以这种方法需要较大的局部窗口, 这会带来局部窗口可能超越纹理区域、边界多检测等一系列问题。3) 从图 3(c) 可以看到, 当分析尺度足够时, 所有的纹理均表现出色彩分布均匀的性质, 而纹理边界则不具备此性质, 因此可以通过均匀度映像上的区域生长来确定各纹理区域。

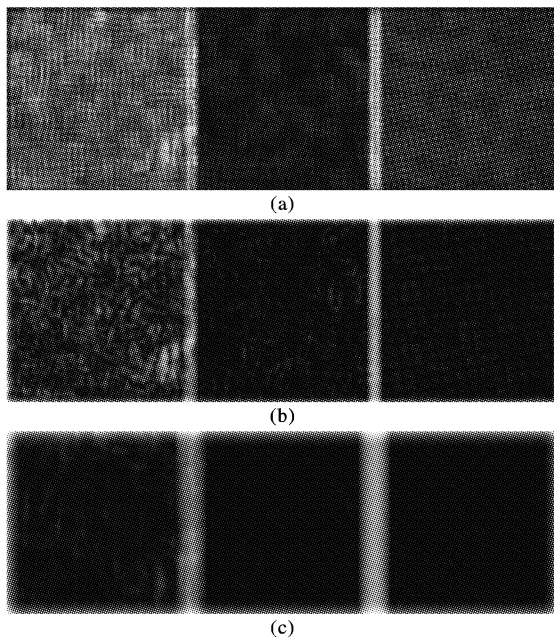


图 3 将图 1 的三个样本拼接后的量化图像及其均匀度映像

另外, 还用前面提到的分割原型系统对大量自然图像进行了分割实验, 图 4 中给出了其中的一个例子: (a) 是大小为 352×288 的原始图像; (b) 是量化后图像, $K = 10$; (c)、(d)、(e) 是在 33 、 17 、 9 三个尺度上计算的均匀度映像; (f) 是最后的分割结果, 区域的边界用白线在原图中描出。从这些结果来看, ISBEC 算法可以有效地将自然图像分割成一些色彩—纹理一致的区域。而且, 如果将算法中的色彩量化替换为标量量化, 或者更简单地直接将灰度图像视为三个色彩分量相同的彩色图像, 则 ISBEC 算法对自然灰度图像分割也将是有效

的。

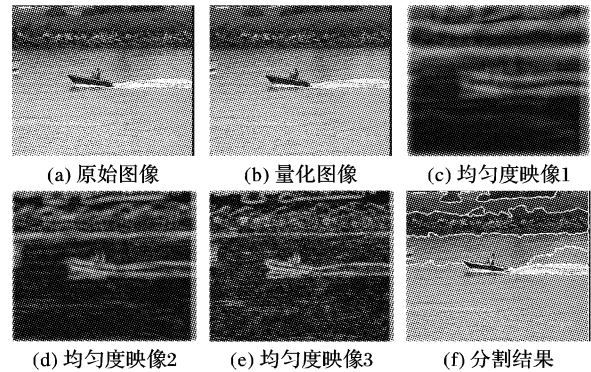


图 4 ISBEC 对一幅彩色自然图像的分割结果

参考文献:

- [1] LUCCHESI L, MITRA SK. Color Image Segmentation: A State-of-the-art Survey [A]. Proceedings Of the Indian National Science Academy (INSA-A) [C]. Delhi, India, 2001. 207 - 221.
- [2] 章毓晋. 图像分割 [M]. 北京: 科学出版社, 2001. 2 - 3.
- [3] MARTIN DR, FOWLKES CC, MALIK J. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues [J], IEEE Transactions On PAMI, 2004, 26(5): 530 - 549.
- [4] TUCERYAN M, JAIN AK. Texture Analysis [A]. Handbook Pattern Recognition and Computer Vision [C]. 2nd edition. Singapore: World Scientific, 1993. 235 - 276.
- [5] CHEN JQ, PAPPAS TN, MOJSILOVIC A, et al. Adaptive Image Segmentation Based on Color and Texture [A]. Proceedings of ICIP2002 [C]. 2002, Vol. 3. 777 - 780.
- [6] VERTAN C, BOUJEMAA N. Color Texture Classification by Normalized Color Space Representation [A]. Proceedings of ICPR2000 [C]. 2000, Vol 3. 580 - 583.
- [7] VANDENBROUCKE N, MACAIRE L, POSTAIRE JG. Unsupervised Color Texture Feature Extraction and Selection for Soccer Image Segmentation [A]. Proceedings of ICIP2000 [C]. 2000, Vol. 2. 800 - 803.
- [8] JAIN R, KASTURI R, SCHUNCK BG. 机器视觉 [M]. 英文版. 北京, 机械工业出版社, 2003. 234 - 234.
- [9] 王润生. 图像理解 [M]. 长沙: 国防科技大学出版社, 1995. 145 - 146.
- [10] ZHANG X, SONG ZM, WANG YL, et al. Color Quantization of Digital images [A]. Proceedings of PCM2005, LNCS 3768 [C]. 2005. 653 - 664.
- [11] PUZICHA J, RUBNER Y, TOMASI C, et al. Empirical Evaluation of Dissimilarity Measures for Color and Texture [A]. Proceedings of ICCV [C]. 1999.

(上接第 1865 页)

服务在不同用户设备环境的应用提供了很好的灵活性和适应能力。

参考文献:

- [1] DAN A, SITARAM D, SHAHABUDDIN P. Dynamic Batching Policies for an On - Demand Video Server [J]. ACM Multimedia Systems, 1996, (4): 112 - 121.
- [2] VISWANATHAN S, IMIELINSKI T. Pyramid Broadcasting for Video on Demand Service [A]. IEEE Multimedia Computing and Networking Conference [C]. San Jose, California, 1995, Vol 2417. 66 - 77.
- [3] MA HS, TO TPJ. Mirrored-pyramid broadcasting for metropolitan on-demand video delivery [A]. International Workshop on Multimedia

Data Storage, Retrieval, Integration and Applications [C]. 2000. 104 - 109.

- [4] MA HS, TO TPJ, LUN DPK. Enhanced mirrored pyramid broadcasting for video-on-demand delivery [A]. Proceedings of the 2000 IEEE Asian Pacific Conference on CAS [C]. 2000. 875 - 878.
- [5] ZHANG Z, YU S, CHEN J, et al. Advanced mirrored pyramid broadcasting for video-on-demand [J]. IEEE Transactions on Consumer Electronics, 2004, 50(1): 139 - 144.
- [6] LU H, LAN J, ZHUANG X. Advanced pyramid broadcasting for video-on-demand [A]. IEEE Sixth International Symposium on Multimedia Software Engineering (MSE) [C]. 2004.