

一种基于精确度的降分辨率视频转码 运动矢量合成算法

韩 镇¹⁾ 胡瑞敏^{1), 2)} 陈 皓²⁾ 李 明²⁾

¹⁾ (武汉大学计算机学院, 武汉 430072) ²⁾ (武汉大学国家多媒体软件工程技术研究中心, 武汉 430072)

摘 要 针对现有的降分辨率视频转码运动矢量合成算法没有考虑输入运动矢量的精确度, 从而使得合成误差较高的问题, 提出了一种基于精确度的降分辨率视频转码运动矢量合成算法。该算法分为两次合成, 首先基于绝对差值和 (SAD) 对转码输入流中的运动矢量进行平均合成, 然后再基于率失真函数, 对转码输出流中的相邻运动矢量和第 1 次合成得到的运动矢量进行选择合成。仿真结果表明, 与现有算法相比, 该算法在保持较低计算复杂度的情况下, 显著降低了运动矢量合成的误差。

关键词 降分辨率转码 运动矢量合成 精确度 绝对差值和 率失真函数

中图分类号: TN919.81 **文献标志码:** A **文章编号:** 1006-8961(2010)01-0007-08

Motion Vector Composition Algorithm Based on Accuracy for Reduced Spatial Resolution Video Transcoding

HAN Zhen¹⁾, HU Ruimin^{1), 2)}, CHEN Hao²⁾, LIMing²⁾

¹⁾ (Computer School, Wuhan University, Wuhan 430072)

²⁾ (National Multimedia Software Engineering Research Center, Wuhan University, Wuhan 430072)

Abstract The current motion vector composition algorithms for reduced spatial resolution video transcoding does not consider the accuracy of input motion vectors. So the errors of these algorithms are rather big. In order to solve this problem, this paper proposes a motion vector composition algorithm based on the accuracy for reduced spatial resolution video transcoding. This algorithm includes two compositions. Firstly, it composes the input motion vectors by averaging them based on SAD. Secondly, it composes the neighboring motion vectors and composed motion vector in the first composition by selecting them based on rate distortion function. The simulation results show that the proposed algorithm reduces the error compared with the algorithms while maintaining the low complexity.

Keywords reduced spatial resolution transcoding motion vector composition accuracy sum absolute difference (SAD), rate distortion function

0 引 言

视频转码能够将已压缩的视频流从一种格式 (如分辨率、帧率、码率等) 转换为另一种格式, 以适应不同性能的网络和终端^[1-2]。这对于解决异构网

络下的通用多媒体存取问题具有重要意义。作为视频转码的关键技术之一, 降分辨率视频转码能够通过空间下采样来将压缩视频由高分辨率转换为低分辨率, 以适应终端显示能力和处理能力的限制, 其在移动娱乐、移动监控等领域具有广泛的应用价值。由于在降分辨率转码中, 高分辨率图像中的多个宏块需

基金项目: 国家自然科学基金项目 (60472040); 公安部应用创新计划项目 (2005YYCXHBS114)

收稿日期: 2008-05-09 改回日期: 2008-12-05

第一作者简介: 韩镇 (1980—), 男, 2009 年获武汉大学计算机学院博士学位, 现为该校老师。主要研究领域为视频编码和转码。E-mail

hanzhen_1980@163.com

要合成为低分辨率图像中的一个宏块,相应地多个运动矢量也需要合成一个新的运动矢量,因此运动矢量合成成为降分辨率视频转码的关键问题^[1-2]。

为了更具一般性,本文将运动矢量合成定义为多个输入运动矢量到一个输出运动矢量的映射。根据这一定义,已有的运动矢量合成算法可以总结为以下两种映射:第 1 种映射是选择,第 2 种映射是平均或求和。根据第 1 种映射思路,文献[3]和文献[4]分别提出了中值法、空间主导向量选择等合成算法,即从多个输入运动矢量中选择具有中值和最大覆盖面积的运动矢量作为输出运动矢量。根据第 2 种映射思路,文献[5-7]分别提出了基于活动性、基于编码复杂度和基于面积的合成算法,即将多个输入运动矢量,根据面积、活动性和编码复杂度进行加权平均来得到输出运动矢量;而文献[8]则提出了基于线性衰退模型的合成算法,该算法将多个输入运动矢量根据线性模型系数进行加权求和得到输出运动矢量。

上述算法虽然对运动矢量合成的性能有所改善,但是却存在着一个共同的问题:没有考虑输入运动矢量的精确度,即假设输入的运动矢量是足够精确的。但是在降分辨率视频转码中,这一假设并不一定成立,因为首先,当与输入的运动矢量对应的采样面积小于搜索面积(例如当搜索面积为 16×16 而采样面积为 4×4)时,输入的运动矢量相对于搜索面积虽然足够精确,但是相对于采样面积则不一定是足够精确的;其次,当转码前后的视频流所采用的运动矢量的精确度量不同(例如当转码前输入的 MPEG-2 视频流和转码后输出的 H. 264 视频流分别采用绝对差值和 (SAD) 和率失真函数作为运动矢量精确度的度量)时,输入的运动矢量相对于输入的视频流虽然足够精确,但是相对于输出的视频流则不一定是足够精确的。

针对这一问题,本文提出了一种基于精确度的降分辨率视频转码运动矢量合成算法。该算法分为两次合成,即首先基于 SAD 对转码输入流中的运动矢量进行平均合成;然后再基于率失真函数对转码输出流中的相邻运动矢量和第 1 次合成得到的运动矢量进行选择合成。

1 基于 SAD 的第 1 次合成

在第 1 次运动矢量合成中,本文算法采用 SAD

作为运动矢量的精确度量,基于 SAD 对转码输入流中的多个运动矢量进行加权平均。正如上文所分析的,在降分辨率视频转码中,由于转码输入流中的运动矢量不一定是足够精确的,而 SAD 对于不同精确度的运动矢量又具有不同的意义,因此在对运动矢量进行基于 SAD 的加权平均时,需要根据其精确度的高低分别进行讨论。

当转码输入流中的运动矢量精确度较低时,由于 SAD 还未达到最小值, SAD 越大,意味着运动矢量的精确度越低,因此在合成时,运动矢量的权重也就越小。在这种情况下,运动矢量的权重应当与 SAD 成反比关系。

当转码输入流中的运动矢量精确度较高时,由于 SAD 已达到或接近最小值, SAD 越大,意味着输入运动矢量对应块的空间活动性越高,因此在合成时,输入运动矢量的权重也就越大。在这种情况下,运动矢量的权重应当与 SAD 成正比关系。

当转码输入流中的运动矢量精确度介于上述两种情况之间时,运动矢量的权重也应当介于上述两种情况之间。

基于上述分析,3 种情况下的权重计算公式分别为

$$W_i^{\text{low}} = S_i / \frac{SAD_i}{S_i} = S_i^2 / SAD_i \quad (1)$$

$$W_i^{\text{high}} = S_i \cdot \frac{SAD_i}{S_i} = SAD_i \quad (2)$$

$$W_i^{\text{midling}} = S_i \quad (3)$$

其中, W_i^{low} , W_i^{high} 和 W_i^{midling} 分别表示第 i 个输入运动矢量 V_i^{motion} 在精确度较低、精确度较高和介于两种情况之间时的权重, S_i 和 SAD_i 分别表示与 V_i^{motion} 对应的采样面积和 SAD 值, SAD_i / S_i 表示单位采样面积的 SAD 值。 S_i 可以通过采样比例计算得到,而 SAD_i 则可以通过解码转码输入视频流得到。

根据上述不同的权重对转码输入流中的运动矢量进行加权平均,即可以得到不同情况下的合成运动矢量。当多个输入运动矢量精确度均较低、均较高以及均介于上述两种情况之间时,合成运动矢量分别如下所示:

$$V^{\text{low}} = \frac{\sum_{i=0}^n (S_i^2 / SAD_i \cdot V_i^{\text{motion}})}{\sum_{i=0}^n (S_i^2 / SAD_i)} \quad (4)$$

$$V^{high} = \frac{\sum_{i=0}^n (SAD_i \cdot V_i^{motion})}{\sum_{i=0}^n SAD_i} \quad (5)$$

$$V^{midling} = \frac{\sum_{i=0}^n (S_i \cdot V_i^{motion})}{\sum_{i=0}^n S_i} \quad (6)$$

其中, V^{low} , V^{high} , $V^{midling}$ 分别表示转码输入流中的运动矢量在精确度较低、较高和介于两种情况之间时的合成运动矢量, n 表示输入运动矢量的数目。

上述均是针对多个运动矢量的精确度较为一致的情况。下面讨论多个运动矢量的精确度并不一致的情况。导致转码输入流中多个运动矢量精确度不一致的主要原因是与运动矢量对应的采样面积与搜索面积的比例关系不一致的缘故: 当运动矢量对应的采样面积越接近搜索面积时, 运动矢量的精确度就越高; 反之, 运动矢量的精确度就越低。因此当转码输入流中多个运动矢量的精确度不一致时, 可以首先根据与各运动矢量对应的采样面积与搜索面积的比例关系来判断其精确度的高低, 然后根据各运动矢量的精确度高低选择对应的权重来进行加权平均。因此在这种情况下, 权重和合成运动矢量的计算公式分别为

$$W_i^{m \times} = \begin{cases} S_i^2 / SAD_i & \text{当 } S_i / S < 1/4 \text{ 时} \\ SAD_i & \text{当 } S_i / S > 3/4 \text{ 时} \\ S_i & \text{其他} \end{cases} \quad (7)$$

$$V^{m \times} = \frac{\sum_{i=0}^n (D_i \cdot V_i^{motion})}{\sum_{i=0}^n D_i} \quad (8)$$

其中, $W_i^{m \times}$ 表示输入的第 i 个运动矢量 V_i^{motion} 在精确度不一致时的权重, S_i 表示与第 i 个运动矢量 V_i^{motion} 对应的搜索面积, $V^{m \times}$ 表示输入运动矢量在精确度不一致时的合成矢量。

但是需要特别说明的是: 在第 1 次合成中, 本算法对每组 (n 个) 输入运动矢量不论精确度的高低均根据式 (4) ~ 式 (6) 式 (8) 进行 4 次平均合成, 因此并不需要确定输入运动矢量的精确度。

2 基于率失真函数的第 2 次合成

在第 1 次合成中, 本文算法基于 SAD, 对转码输

入流中的运动矢量进行了不同的加权平均, 尽管得到 4 个不同的合成运动矢量, 但是这 4 个合成运动矢量的精确度还无法确定。而且如果 4 个合成运动矢量的精确度均较低时, 还需要参考转码输出流中的相邻运动矢量。因此在第 2 次合成中, 本文算法基于率失真函数对转码输出流中的相邻运动矢量和第 1 次合成得到的 4 个运动矢量进行选择合成, 选择精确度最高, 即具有最小率失真函数值的运动矢量为输出的合成矢量。

转码输出流中的相邻运动矢量包括中值运动矢量 V^{medium} 和最远运动矢量 V^{far} 。所谓最远运动矢量是指左、上、左上、右上 4 个相邻运动矢量中距离第 1 次合成矢量最远的运动矢量。其中每个相邻运动矢量与第 1 次合成矢量之间的距离由该相邻运动矢量与 4 个合成矢量之间距离的最小值决定。之所以引入最远运动矢量是因为在第 2 次运动矢量合成中, 要考虑转码输出流中相邻运动矢量, 其目的就在于增强候选运动矢量的多样性, 使得在第 1 次合成矢量的精确度较低时, 也能通过相邻运动矢量获得精确度较高的合成矢量。由于在第 1 次合成矢量的精确度较低时, 相邻运动矢量与其距离越近, 精确度较高的可能性就越小; 反之, 精确度较高的可能性就越大。由于选择距离其最远的运动矢量作为候选运动矢量, 可以增强候选运动矢量的多样性, 从而使得获得精确度较高的合成矢量的可能性增大。

在第 2 次合成中, 率失真函数如下所示:

$$J(m) = SAD_m + \lambda_{motion} R_m \quad (9)$$

其中, m 表示当前块的运动矢量 (候选范围包括转码输出流中的相邻运动矢量 V^{medium} , V^{far} 和第 1 次合成得到的运动矢量 V^{low} , V^{high} , $V^{midling}$, $V^{m \times}$); $J(m)$ 表示当前块运动矢量为 m 时的率失真函数值; SAD_m 表示当前块运动矢量为 m 时的绝对差值和, 即失真函数; R_m 表示当前块运动矢量为 m 时运动矢量差值的编码比特数, 即率函数; λ_{motion} 表示拉格朗日乘子。下面主要讨论 SAD_m , R_m 的计算。

2.1 失真函数

在第 1 次合成中, 对于转码输入流中的运动矢量, 可以通过解码转码输入视频流来得到其对应的 SAD 值。在第 2 次合成中, 由于当前块还没有进行实际编码, 因此对于第 1 次合成得到的运动矢量和转码输出流中的相邻运动矢量, 均无法直接得到其对应的 SAD 值。

对于第 1 次合成得到的运动矢量, 其对应的

SAD值可以通过对转码输入流中与运动矢量对应的 SAD值进行加权平均得到。 V^{kw} , V^{high} , V^{middle} 和 V^{low} 对应的失真函数 SAD^{kw} , SAD^{high} , SAD^{middle} 和 SAD^{low} 可分别由以下公式估计得到:

$$SAD^{kw} = \frac{\sum_{i=0}^n S_i^2}{\sum_{i=0}^n (S_i^2 / SAD_i)} \quad (10)$$

$$SAD^{high} = \frac{\sum_{i=0}^n SAD_i^2}{\sum_{i=0}^n SAD_i} \quad (11)$$

$$SAD^{middle} = \frac{\sum_{i=0}^n (S_i \cdot SAD_i)}{\sum_{i=0}^n S_i} \quad (12)$$

$$SAD^{low} = \frac{\sum_{i=0}^n (D_i \cdot SAD_i)}{\sum_{i=0}^n D_i} \quad (13)$$

对于转码输出流中的相邻运动矢量(包括中值运动矢量 V^{middle} 和最远运动矢量 V^{far}),其在当前块中对应的失真函数 SAD^{middle} 和 SAD^{far} 可以根据 SAD的计算公式得到。需要指出的是,当相邻运动矢量不可用时,本文算法以转码输出流中前一帧的对应块的运动矢量来代替。

2.2 率函数

如图 1所示, E 为当前块, G 为后续块。由于当前块 E 的运动矢量同时也是后续块 G 的相邻运动矢量,因此当前块 E 的运动矢量的选择不仅影响当前块 E 的运动矢量差值的编码比特数,而且也会因影响后续块 G 的预测运动矢量而影响运动矢量差值的编码比特数。因此在本文算法中,率函数不仅包括当前块的运动矢量与预测运动矢量之间差值的编码比特数,也包括后续块的运动矢量与预测运动矢量之间差值的编码比特数。

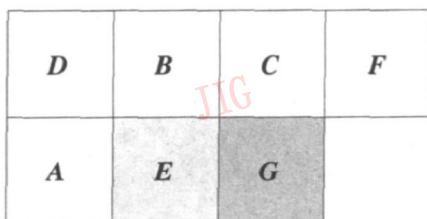


图 1 当前块与后续块位置关系示意图

Fig 1 The positions of current and following block

率函数 R_m 的具体计算公式为

$$R_m = R(m - p) + R(\tilde{m} - \tilde{p}(m)) \quad (14)$$

其中, m 表示当前块的运动矢量, p 表示当前块的预测运动矢量, \tilde{m} 表示后续块的运动矢量, $\tilde{p}(m)$ 表示后续块在当前块运动矢量为 m 时的预测运动矢量。对于当前块运动矢量差值的编码比特数,由于 m 与 p 均已知,因此两者之间差值的编码比特数可以通过查表直接得到。对于后续块的运动矢量差值的编码比特数, $\tilde{p}(m)$ 可以根据当前块的运动矢量 m 以及其他的相邻运动矢量得到, \tilde{m} 可以先用转码输入流中后续块的运动矢量来近似替代,然后通过查表即可得到运动矢量差值的比特数。

3 实验结果

本文基于 H. 264 JM 12.2参考模型和降分辨率转码结构^[8],对运动矢量全搜索算法、文献[8]中的运动矢量合成算法以及本文提出的运动矢量合成算法进行了测试。测试序列使用 Stefan, Foreman, Bus 3个具有不同运动特性的测试序列。转码输入序列的格式为 MPEG-2,量化步长为 16,帧率为 25fps,分辨率为 CIF,转码输出序列的格式为 H. 264,量化步长为 24、28、32、36,帧率与输入序列相同,分辨率为输入序列分辨率的 1/2、2/3。测试序列不使用多参考帧运动预测,均采用 IPPP...的编码结构。运动矢量全搜索算法的搜索窗口为 $[-16, 16]$,文献[8]算法以及本文算法的搜索窗口为 $[-2, 2]$ 。

表 1和表 2分别给出了下采样比率为 1/2和 2/3时,文献[8]算法以及本文算法相对于运动矢量全搜索算法在信噪比、码率、时间 3方面的变化。由表 1和表 2中可以看出,与运动矢量全搜索算法相比,两种算法的信噪比和时间变化基本相同,信噪比均平均下降 0.04dB左右,时间均平均减少 70%左右,而码率的变化差别则较大:下采样比率为 1/2时,文献[8]算法以及本文算法码率分别增加了 12.22%和 1.01%;下采样比率为 2/3时,码率则分别增加了 15.79%和 1.64%。由于码率的增加主要是由合成误差导致的,因此由表 1和表 2可知,本文算法与文献[8]算法相比,复杂度基本保持不变,而合成误差则显著降低。

表 1 下采样比率为 1/2 时合成算法的性能比较 (MPEG-2 到 H. 264 转码)

Tab 1 Comparison of the performance of composition algorithms when downsizing ratio is 1/2 (MPEG-2 to H. 264 transcoder)

测试序列	量化步长	文献 [8] 算法			本文算法		
		信噪比变化 /dB	码率变化 %	时间变化 %	信噪比变化 /dB	码率变化 %	时间变化 %
Stefan	24	+ 0.02	+ 16.46	- 68.38	0	+ 1.12	- 67.17
	28	- 0.04	+ 21.78	- 72.12	- 0.02	+ 1.91	- 70.38
	32	- 0.06	+ 28.92	- 70.44	- 0.03	+ 1.77	- 69.29
	36	- 0.09	+ 34.97	- 74.13	- 0.05	+ 2.13	- 72.26
Foreman	24	+ 0.02	+ 2.78	- 69.78	0	- 0.06	- 69.35
	28	- 0.01	+ 3.96	- 71.90	- 0.02	- 0.42	- 70.99
	32	- 0.07	+ 5.97	- 70.51	- 0.08	+ 0.40	- 70.07
	36	- 0.15	+ 7.07	- 73.43	- 0.10	+ 0.47	- 72.86
Bus	24	+ 0.01	+ 2.94	- 70.26	- 0.01	+ 0.31	- 67.08
	28	- 0.02	+ 4.99	- 71.19	- 0.01	+ 0.89	- 69.88
	32	- 0.06	+ 7.50	- 73.89	- 0.03	+ 1.60	- 71.96
	36	- 0.07	+ 9.26	- 72.29	- 0.04	+ 1.98	- 70.81
平均		- 0.04	+ 12.22	- 71.53	- 0.03	+ 1.01	- 70.18

表 2 下采样比率为 2/3 时合成算法的性能比较 (MPEG-2 到 H. 264 转码)

Tab 2 Comparison of the performance of composition algorithms when downsizing ratio is 2/3 (MPEG-2 to H. 264 transcoder)

测试序列	量化步长	文献 [8] 算法			本文算法		
		信噪比变化 /dB	码率变化 %	时间变化 %	信噪比变化 /dB	码率变化 %	时间变化 %
Stefan	24	+ 0.01	+ 15.68	- 69.66	- 0.01	+ 1.31	- 67.12
	28	- 0.05	+ 21.29	- 71.03	- 0.02	+ 1.60	- 68.34
	32	- 0.08	+ 30.90	- 71.24	- 0.03	+ 1.80	- 68.49
	36	- 0.11	+ 37.64	- 72.83	- 0.04	+ 1.72	- 70.06
Foreman	24	+ 0.01	+ 3.99	- 69.50	+ 0.01	- 0.22	- 68.44
	28	- 0.03	+ 5.12	- 71.87	- 0.01	- 0.44	- 68.84
	32	- 0.05	+ 6.34	- 70.54	- 0.04	+ 1.19	- 70.83
	36	- 0.12	+ 7.45	- 72.29	- 0.10	+ 1.16	- 71.61
Bus	24	+ 0.01	+ 7.61	- 69.67	0	+ 1.33	- 67.77
	28	- 0.01	+ 12.08	- 73.93	- 0.01	+ 2.03	- 71.29
	32	- 0.02	+ 17.61	- 72.76	- 0.02	+ 3.32	- 70.11
	36	- 0.07	+ 23.81	- 73.91	- 0.10	+ 4.83	- 72.52
平均		- 0.04	+ 15.79	- 71.60	- 0.03	+ 1.64	- 69.62

为了进一步说明不同算法的性能,图 2 和图 3 分别给出了 Stefan 和 Foreman 序列在下采样比率为 1/2 和 2/3 时,运动矢量全搜索算法、文献 [8] 算法以及本文算法的率失真曲线。由图 2 和图 3 可以看出,在相

同码率下,本文算法的信噪比始终高于文献 [8] 算法,其中对于运动复杂的 Stefan 序列,信噪比的增益较大,约有 0.8~1.4 dB 的增益;对于运动较少的 Foreman 序列,信噪比的增益较小,约有 0.2~0.4 dB 的增益。

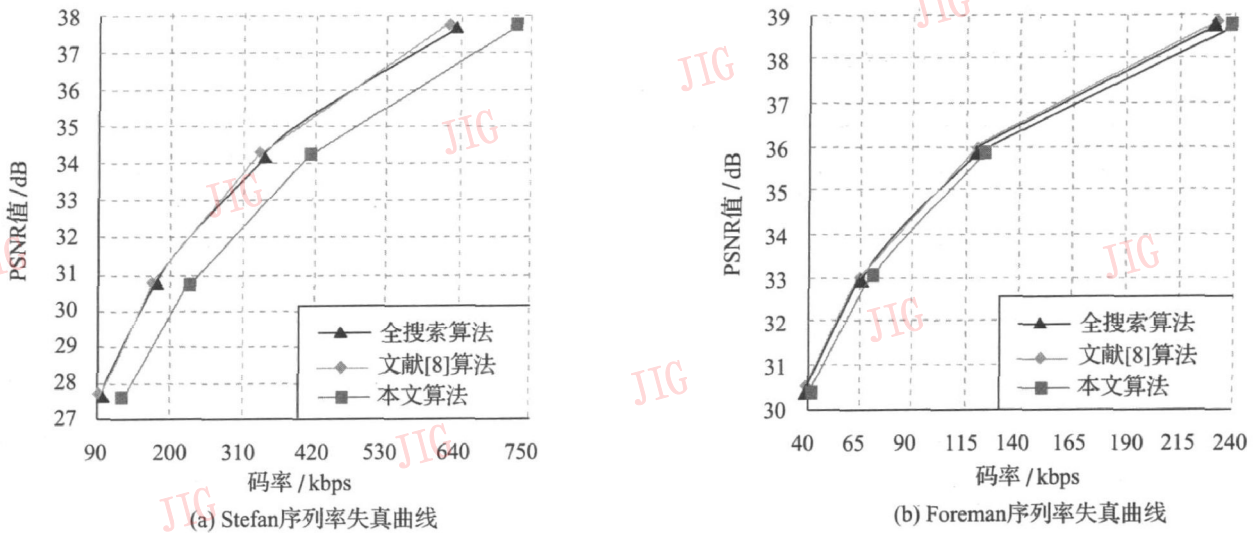


图 2 下采样比率为 1/2 时的率失真曲线示意图

Fig. 2 Schematic of the R-D curves of sequences when downsizing ratio is 1/2

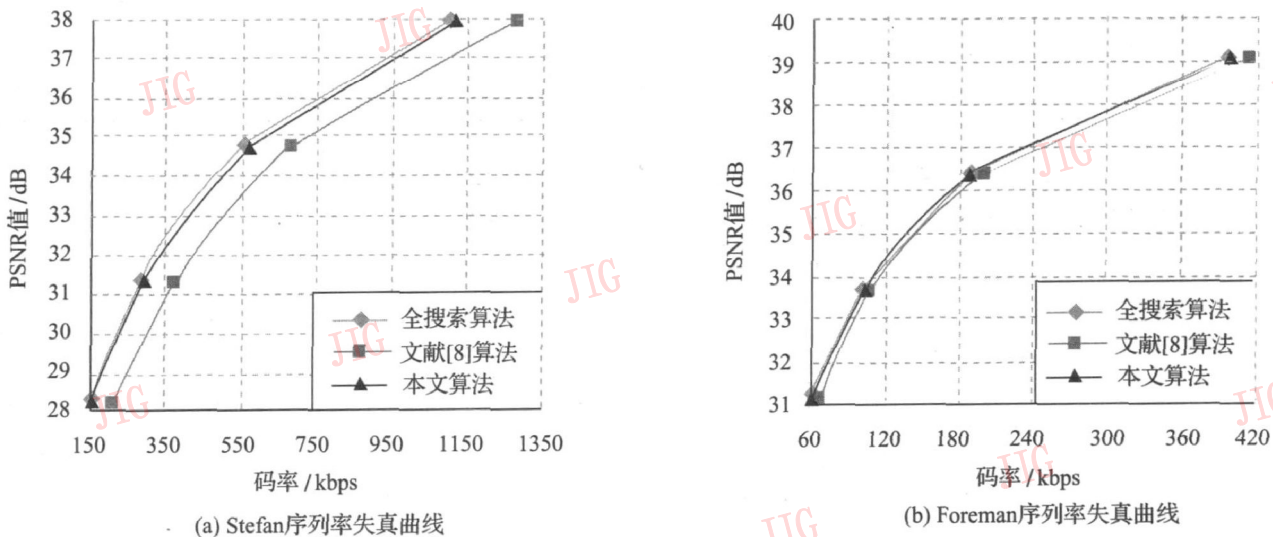


图 3 下采样比率为 2/3 时的率失真曲线示意图

Fig. 3 Schematic of the R-D curves of sequences when downsizing ratio is 2/3

为了进一步说明算法的普适性, 本文还在 M PEG-2 到中国制定的信源编码标准 AVS (audio video coding standard) 的降分辨率转码平台上对运动矢量全搜索算法、文献 [8] 算法以及本文算法进行了测试。测试条件与 M PEG-2 到 H. 264 转码基本相同, 只是量化步长分别为 28、32、36、40。表 3 给出了下采样比率为 1/2 时, 文献 [8] 算法以及本文算法相对于运动矢量全搜索算法在信噪比、码率、时间 3 方面的变化。图 4 给出了 Stefan 和 Foreman 序列在下采样比率为 1/2 时, 3 种算法的率失真曲线。由表 3 和图 4 可以看出, 本算法的复杂度基本

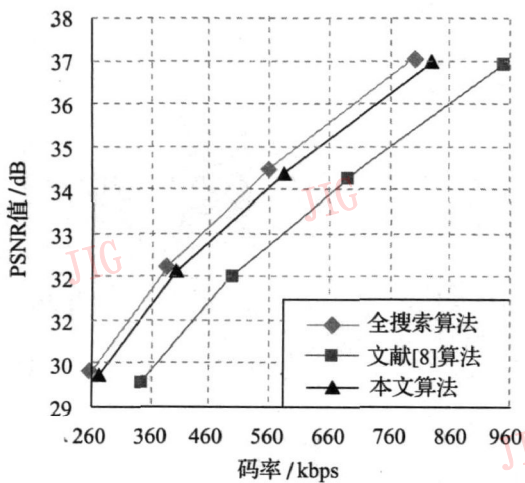
保持不变, 而合成误差则显著降低。

本文算法相对于文献 [8] 算法的增益主要来源于对输入运动矢量精确度的考虑: 由于在第 1 次合成中基于精确度对运动矢量进行了加权平均, 使得运动矢量的精确度越低, 其权重也越低, 从而使得平均合成的运动矢量误差减小; 由于在第 2 次合成中基于精确度对运动矢量进行了率失真选择, 即从第 1 次合成得到的运动矢量和相邻运动矢量中选择率失真函数值最小的作为合成运动矢量, 从而使得选择合成的运动矢量误差减小。

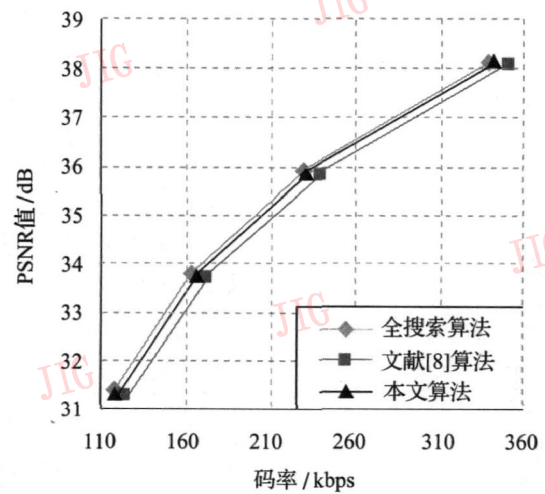
表 3 下采样比率为 1/2 时合成算法的性能比较 (MPEG-2 到 AVS 转码)

Tab 3 Comparison of the performance of composition algorithms when downsizing ratio is 1/2 (MPEG-2 to AVS transcoder)

测试序列	量化步长	文献 [8] 算法			本文算法		
		信噪比变化 /dB	码率变化 %	时间变化 %	信噪比变化 /dB	码率变化 %	时间变化 %
Stefan	28	- 0.13	+ 18.95	- 52.52	- 0.06	+ 3.25	- 48.48
	32	- 0.17	+ 24.30	- 50.52	- 0.07	+ 4.52	- 47.92
	36	- 0.21	+ 28.68	- 48.89	- 0.07	+ 4.31	- 44.64
	40	- 0.26	+ 33.53	- 46.51	- 0.09	+ 5.18	- 43.02
Foreman	28	- 0.05	+ 3.96	- 52.04	- 0.03	+ 1.48	- 51.02
	32	- 0.06	+ 4.85	- 48.91	- 0.03	+ 1.88	- 47.83
	36	- 0.10	+ 5.94	- 47.73	- 0.05	+ 2.14	- 45.45
	40	- 0.08	+ 5.95	- 46.67	- 0.03	+ 2.55	- 45.44
Bus	28	- 0.03	+ 3.89	- 57.41	- 0.01	+ 0.63	- 55.56
	32	- 0.04	+ 4.18	- 55.78	- 0.02	+ 0.79	- 53.85
	36	- 0.05	+ 5.59	- 54.90	- 0.02	+ 1.14	- 54.40
	40	- 0.07	+ 7.79	- 53.06	- 0.02	+ 1.29	- 51.02
平均		- 0.10	+ 12.17	- 51.42	- 0.04	+ 2.42	- 49.92



(a) Stefan 序列率失真曲线



(b) Foreman 序列率失真曲线

图 4 下采样比率为 1/2 时的率失真曲线示意图

Fig 4 Schematic of the R-D curves of sequences when downsizing ratio is 1/2

4 结 论

本文提出了一种基于精确度的降分辨率视频转码运动矢量合成算法。该算法分为两次合成,即分别基于 SAD 和率失真函数对转码输入流中的运动矢量进行平均合成和选择合成。仿真结果表明,与

现有算法相比,该算法在保持较低计算复杂度的情况下,显著降低了运动矢量合成的误差。

参考文献 (References)

[1] Ahmad Ishaq Wei Xiao-hui Sun Yu et al Video transcoding An overview of various techniques and research issues [J]. IEEE

- Transactions on Multimedia, 2005, 7(5): 791-804
- [2] Xin Jun, Lin Chia-wen, Sun Ming-ting. Digital video transcoding [J]. Proceedings of the IEEE, 2005, 93(1): 84-97.
- [3] Xin Jun, Sun Ming-ting, Choi Byung-sun, et al. An HDTV-to-SDTV spatial transcoder [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(1): 998-1008.
- [4] Kumar Rajeev, Patil Vasant. An efficient motion vector composition scheme for arbitrary frame down-sampling video transcoder [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(9): 1148-1152.
- [5] Liang Yong-qing, Chau Lap-pui, Tan Yap-peng. Arbitrary downsizing video transcoding using fast motion vector reestimation [J]. IEEE Signal Processing Letters, 2002, 9(11): 352-355.
- [6] Tan Yap-peng, Liang Yong-qing, Sun Hai-wei. On the methods and performances of rational downsizing video transcoding [J]. Signal Processing: Image Communication, 2004, 19(1): 47~65.
- [7] Tan Yap-peng, Sun Hai-wei. Fast motion re-estimation for arbitrary downsizing video transcoding using H. 264/AVC standard [J]. IEEE Transactions on Consumer Electronics, 2004, 50(3): 887-894.
- [8] Wang Jiao, Yang En-hui, Yu Xiang. An efficient motion estimation method for H. 264-based video transcoding with spatial resolution conversion [C] // Shi Yun-Qing. Proceedings of 2007 IEEE International Conference on Multimedia and Expo. Piscata, NJ, USA: IEEE Computer Society, 2007: 444-447.