

# 一种用于 MPEG-4 形状编码的快速运动估计算法<sup>1</sup>

卢官明

(南京邮电学院信息工程系 南京 210003)

**摘 要** 该文提出了一种用于 MPEG-4 形状编码的快速运动估计算法。该算法利用了形状编码及形状信息的固有特性,即基于上下文的运动估计特性、相邻二值 alpha 块的运动矢量相关性以及形状信息的二值特性。模拟结果表明,该算法具有运算量少、处理速度快的特点,适用于 MPEG-4 形状编码的实时软件实现。

**关键词** 运动估计, 形状编码, MPEG-4, 基于对象编码

**中图分类号** TN919.8

## 1 引 言

MPEG-4 是由活动图像专家组 (MPEG, Moving Pictures Expert Group) 制定的有关多媒体通信的国际标准,它提供了用于对自然 / 合成的视频 / 音频进行基于对象编码 (object-based coding) 的工具集<sup>[1]</sup>。所谓对象就是在一个场景中能够进行访问 (搜索和浏览) 和操作 (剪贴) 的实体。基于对象的编码除了能提高数据压缩比外,还能实现许多基于内容 (content-based) 的交互功能。MPEG-4 通过引入视频对象面 (VOP, Video Object Plane) 的概念支持基于内容的功能,在多媒体通信、多媒体存储等领域具有广泛的应用,包括低比特率的多媒体通信,如视频电话、视频电子邮件、移动多媒体通信、电子新闻等; Internet 上的视频流与可视游戏,如网上电影; 实时多媒体远程监控; 基于内容存储和检索的多媒体系统; 基于面部表情模拟的虚拟会议; DVD 上的交互多媒体应用,等等。由于 VOP 的形状通常是任意的,所以,在 MPEG-4 视频编码中需要进行形状编码。

为了提高编码效率,运动估计是必不可少的。然而,运动估计的运算量在形状编码及纹理编码中占据极大的比例,因此,对实时 MPEG-4 编码器来说是重大的负担。近几年来,已提出了不少快速运动估计算法,但大都适用于纹理编码。本文提出的快速运动估计算法则适用于形状编码。

本文的组织是这样的:首先介绍在 MPEG-4 视频验证模型 (VM, Verification Model) 中采用的用于形状编码的运动估计方法;然后,再具体描述本文提出的快速运动估计算法;最后,给出仿真结果及结论。

## 2 MPEG-4 验证模型中采用的运动估计方法

MPEG-4 引入了形状信息的编码,尽管形状编码在计算机图形学、计算机视觉和图像压缩领域不是什么新技术,但将其纳入完整的视频编码标准内,这还是第一次<sup>[2]</sup>。VOP 的形状信息有两类:二值形状信息和灰度形状信息。二值形状信息用 0, 1 来表示 VOP 的形状,0 表示该像素在 VOP 区域之外,1 表示在 VOP 区域之内,对二值形状信息进行编码时,采用基于块的运动补偿技术,可以是有损编码,也可以是无损编码。灰度形状信息用 0 ~ 255 之间的数值来表示 VOP 内各像素的透明度,其中 0 表示完全透明 (相当于二值形状信息中的 0),255 表示完全不透明 (相当于二值形状信息中的 1)。灰度形状信息的编码采用基于块的运动补偿 DCT 方法 (同纹理编码相似),属于有损编码。目前的标准中采用矩阵的形式来表示二值或灰度形状信息,称之为位图 (或 alpha 平面)。实验表明,位图表示法具有较高的编码效率和较低的运算复杂度。

<sup>1</sup> 2002-03-15 收到, 2002-10-09 改回

在 MPEG-4 验证模型中, VOP 的形状编码是基于  $16 \times 16$  像素的 BAB(Binary Alpha Block, 二值 alpha 块)。如果该 VOP 的编码模式是 B-VOP(帧间双向预测编码模式) 或 P-VOP(帧间预测编码模式), 则要对待编码的 BAB 进行运动估计, 得到用于形状编码的运动矢量 (MVs, Motion Vector for shape)。MVs 可表示成用于形状编码的运动矢量预测值 (MVPs) 与用于形状编码的运动矢量差值 (MVDs) 之和。

### 2.1 用于形状编码的运动矢量预测值 (MVPs)

当前 BAB 的 MVs 可用相邻 BAB 的 MVs 值来预测, 即用当前 BAB 的左、上和右上边的 BAB 的 MVs1, MVs2 和 MVs3 作为候选的 MVPs, 如图 1 所示。通过比较 MVPs 所指向的参考 BAB 与当前 BAB 来计算运动补偿 (MC, Motion Compensation) 误差, 通常用差值的绝对值之和 (SAD, Sum of Absolute Difference) 值来度量。其中使 SAD 值为最小的那个候选运动矢量则作为 MVPs。

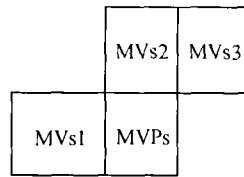


图 1 当前 BAB 的 MVPs 可从相邻 BAB 的 MVs 中选取

### 2.2 用于形状编码的运动矢量 (MVs)

在求得 MVPs 的基础上, 再按照下列步骤来计算 MVs:

(1) 将  $16 \times 16$  的 BAB 分为 16 个  $4 \times 4$  的子块, 对于任一  $4 \times 4$  子块, 如果计算的 MC 误差小于或等于一个指定的阈值 ( $16 \times \text{alpha-th}$ ), 则 MVPs 可直接用作 MVs, 并终止求解 MVs 的过程。

(2) 如果上述条件不满足, 则在预测矢量 MVPs 附近搜索运动矢量 (MV)。搜索在以 MVPs 为中心沿水平和垂直方向最大为  $\pm 16$  像素的范围内进行, 并采用全搜索方法, 使 SAD 最小的那个 MV 取作 MVs, 并进而求得用于形状编码的运动矢量差值 (MVDs), 即  $\text{MVDs} = \text{MVs} - \text{MVPs}$ 。

如果有多个 MVs 使得 SAD 有相同的最小值, 则选择其中使 MVDs 的码字长度最短的那个 MVDs。如果有多个 MVs 使得 SAD 有相同的最小值, 并且使得 MVDs 的码字长度也相同, 则选择具有较小垂直分量的 MVDs; 如果 MVDs 的垂直分量也相同, 则选择具有较小水平分量的 MVDs。

## 3 提出的快速运动估计算法

图 2 给出了提出的快速运动估计算法的流程图。为了加快运动估计的处理速度, 在图 2 中通过对 MC 误差与给定阈值 (THR) 的比较给出了两个跳转机会。这里, 值得注意的是, THR 不同于  $\text{alpha-th}$ 。在比较中, 我们利用了 3 个特性, 即基于上下文的运动估计特性、相邻 BAB 运动矢量之间的相关性以及形状信息的二值特性。

### 3.1 基于上下文的运动估计特性

通常, 用于纹理编码的运动估计是基于 SAD。然而, 用于形状编码的运动估计是基于上下文。MPEG-4 的形状编码采用的是基于上下文的算术编码 (CAE, Context-based Arithmetic Encoding)。因此, 一个提供最小 MC 误差的 MV 并不总能保证有最好的 CAE 性能。所以, 我们使用下列的统计特性来提高运动估计的处理速度: 如果对应于一个 MV 的 MC 误差小于一个给定的 THR, 则执行跳转, 终止对当前 BAB 的运动估计不会影响 CAE 的性能。

在文献 [1] 所描述的方法中, 使用 MVPs 作为初始的 MVs 候选值。如果对应于当前 BAB 的 MVPs 的 MC 误差小于 THR, 我们将 MVs 设置成 MVPs, 并终止运动估计过程。在实验中, 我们取 THR 为 10, 结果即使在最不利的情形下 (即用 Hall monitor 序列), 大约有 30% BAB 的运动估计因通过这个比较判决执行跳转而被终止。

### 3.2 相邻 BAB 运动矢量之间的相关性

在假设每个视频对象为刚体的条件下, 我们利用相邻 BAB 运动矢量之间的相关性来进一步减小计算量。首先, 我们按照对象轮廓边缘的连接关系, 将形状分为 4 类, 如图 3 所示。这里, 黑色区域代表对象 (前景) 还是背景无关紧要, 重要的是要找出对象轮廓边缘的趋向以便考察相邻 BAB 的 MVs 之间的相关性。图 3(a) 示例说明了当前 BAB 中的对象轮廓边缘与它左边 BAB 中的对象轮廓边缘相连的情形; 图 3(b) 描述了当前 BAB 的 MVs 与其上面 BAB 的 MVs 有可能相似的情形; 图 3(c) 说明了当前 BAB 的 MVs 与其相邻两个 BAB

的 MVs 有很强相关性的情形; 图 3(d) 则描述了当前 BAB 的边缘与其他 BAB 的边缘不相连的情形。对于图 3(a), 3(b) 中的两种情形, 我们设当前 BAB 的初始 MVs 等于与之相连 BAB 的 MVs; 对于图 3(c) 情形, 我们设当前 BAB 的初始 MVs 等于与之相连的两个 BAB 的 MVs 之平均值。

为了提高 CAE 的性能, 我们需要对利用运动矢量相关性预测得到的初始 MVs 作进一步的微调。为了降低搜索的复杂性, 在初始 MVs 周围  $\pm 1$  的范围内进行搜索微调。根据我们所作的实验,  $\pm 1$  的局部搜索足以得到可靠的性能。在这一步骤中, 另外有约 10% BAB 的运动估计会执行跳转而被终止。

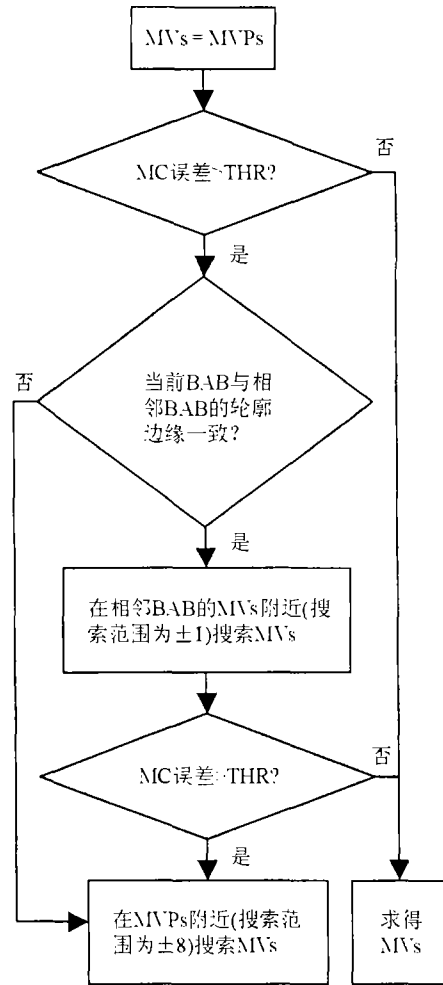


图 2 提出的快速运动估计算法的流程图

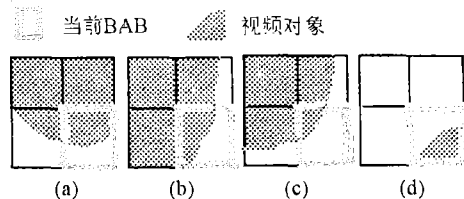


图 3 当前 BAB 中视频对象的 4 类边缘形状

### 3.3 形状信息的二值特性

形状信息用 alpha 平面来描述. alpha 平面有两个值, 即“0”和“1”. “0”表示该像素在 VOP 区域之外, “1”表示在 VOP 区域之内. 我们用这个二值特性来加快运动补偿误差的计算速度. 为了提供形状信息, 一个像素可以用单个比特来表示. 所以, 如果我们用一个变量 (C 语言中的字符类型) 来定义 8 个像素, 则一次就能对 8 个像素进行操作. 换言之, 我们通过对两个 8bit 变量执行“异或” (XOR) 运算并参考预定义的查找表, 就能够计算对应于 8 个像素的 MC 误差. 这样, 我们可以进一步减小计算 MC 误差的运算量.

### 3.4 搜索算法

如果一个 BAB 的运动估计在经历两次比较判决后未被终止, 则我们必须在 MVPs 附近搜索 MVs. 在这一步, 我们可以采用已有的各种搜索算法, 包括三步搜索 (three-step search) 等快速搜索法, 甚至全搜索 (full search) 算法. 本文采用了文献 [3] 中提出的 BBGDS (Block-Based Gradient Descent Search) 快速搜索算法.

## 4 仿真结果

在实验中, 我们使用了 4 个 QCIF(176×144) 格式的 MPEG-4 测试视频序列: Children-Kids, Hall monitor, Akiyo, Container ship. 每个视频测试序列包含 200 个任意形状的 VOP. 对所有的视频测试序列, 在 10Hz 和 30Hz 两种帧频下, 执行无损形状编码. 表中的 #MB 表示不执行跳转的宏块 (BAB) 数. 应注意的是: 在视频验证模型中也可以采用各种快速搜索法. #S-bits/VOP 表示用来描述每个 VOP 形状的平均比特数; #SP 表示在执行运动估计时参与计算的搜索点数, 运动估计的计算量与之成正比. 表中也给出了本文提出的算法相对于视频验证模型中采用的算法所得结果的百分比. 在 Akiyo 和 Container ship 序列中只有很少量的 BAB 需要进行运动矢量搜索. 即使在最不利的情形下 (对于 Hall monitor 序列), 无论采用哪种搜索算法, 也只有约 42% (30Hz 帧频下)~62%(10Hz 帧频下) 的 BAB 需要进行运动矢量搜索. 所以, 该算法具有运算量少、处理速度快的特点, 适用于 MPEG-4 形状编码的实时软件实现.

表 1 (a) 仿真结果 (30Hz 帧频)

序列		全搜索					
		#MB	(%)	#S-Bits/VOP	(%)	#SP	(%)
Children-Kids	VM 中的算法	10332		983		2985948	
	本文算法	2896	28.03	990	100.71	836944	28.03
Akiyo	VM 中的算法	1760		157		508640	
	本文算法	1	0.06	155	98.73	289	0.06
Container ship	VM 中的算法	1966		198		568174	
	本文算法	10	0.51	195	98.48	2890	0.51
Hall monitor	VM 中的算法	2903		278		838967	
	本文算法	1228	42.30	276	99.28	354892	42.30

表 1 (b) 仿真结果 (30Hz 帧频)

序列		BBGDS					
		#MB	(%)	#S-Bits/VOP	(%)	#SP	(%)
Children-Kids	VM 中的算法	10332		990		122648	
	本文算法	2916	28.22	999	100.91	44449	36.24
Akiyo	VM 中的算法	1741		153		16769	
	本文算法	2	0.11	155	101.31	65	0.39
Container ship	VM 中的算法	1934		196		19024	
	本文算法	10	0.52	193	98.47	91	0.48
Hall monitor	VM 中的算法	2900		275		35986	
	本文算法	1226	42.28	276	100.36	15941	44.30

表 2 (a) 仿真结果 (10Hz 帧频)

序列		全搜索					
		#MB	(%)	#S-Bits/VOP	(%)	#SP	(%)
Children-Kids	VM 中的算法	3798		1261		1097622	
	本文算法	1762	46.39	1264	100.24	509218	46.39
Akiyo	VM 中的算法	983		205		284087	
	本文算法	2	0.20	201	98.05	578	0.20
Container ship	VM 中的算法	1281		295		370209	
	本文算法	18	1.41	286	96.95	5202	1.41
Hall monitor	VM 中的算法	974		306		281486	
	本文算法	603	61.91	308	100.65	174267	61.91

表 2 (b) 仿真结果 (10Hz 帧频)

序列		BBGDS					
		#MB	(%)	#S-Bits/VOP	(%)	#SP	(%)
Children-Kids	VM 中的算法	3798		1264		55378	
	本文算法	1765	46.47	1269	100.40	32109	57.98
Akiyo	VM 中的算法	970		203		9783	
	本文算法	3	0.31	201	99.01	99	1.01
Container ship	VM 中的算法	1278		293		13261	
	本文算法	18	1.41	285	97.27	175	1.32
Hall monitor	VM 中的算法	975		306		14698	
	本文算法	602	61.74	308	100.65	10014	68.13

## 参 考 文 献

- [1] ISO/IEC JTC1/SC29/WG11, MPEG99/5477, "MPEG-4 Video Verification Model Version 14.2", Dec. 1999.
- [2] 钟玉琢, 王琪, 贺玉文, 编著, 基于对象的多媒体数据压缩编码国际标准——MPEG-4 及其校验模型, 北京, 科学出版社, 2000 年 10 月.
- [3] L. K. Liu, E. Feig, A block-based gradient descent search algorithm for block motion estimation in video coding, IEEE Trans. on Circuits and Systems for Video Technology, 1996, 6(4), 419-422.

A FAST MOTION ESTIMATION ALGORITHM FOR  
MPEG-4 SHAPE CODING

Lu Guanming

(Dept. of Info. Eng., Nanjing University of Posts &amp; Telecom., Nanjing 210003, China)

**Abstract** This paper proposes a fast motion estimation algorithm for MPEG-4 shape coding. The proposed algorithm takes advantage of inherent properties of shape coding as well as shape information, i.e., context-based motion estimation feature, motion vector correlation between neighboring binary alpha blocks, and binary feature of shape information. Simulation results show that the algorithm can reduce the computational complexity of motion estimation for shape coding and achieve a high processing speed. Therefore, the algorithm is suitable for real-time software implementation of MPEG-4 shape coding.

**Key words** Motion estimation, Shape coding, MPEG-4, Object-based coding

卢官明: 男, 1965 年生, 博士, 副教授, 主要研究方向为图像处理与多媒体通信, 在国际学术会议、核心刊物上发表 20 多篇论文.