

# 块运动估计的研究进展

禹晶 苏开娜

(北京工业大学计算机学院, 北京 100022)

**摘要** 运动估计是根据视频序列中时间上相关的信息估计场景或目标的2维运动向量场的过程。运动估计在超分辨率图像复原中的作用是把所有低分辨率观测帧的像素映射到参考帧的相应位置。精确的运动估计是超分辨率图像复原的关键。基于块的模型是超分辨率图像复原中常用的运动估计方法。介绍了块运动估计的概念,概述了4类快速块运动估计的方法,详细描述了第1类方法中几种主要算法的搜索过程,并将多种快速块匹配算法进行了总结比较。

**关键词** 超分辨率图像复原 运动估计 块匹配算法 搜索模式

中图分类号: TP391 文献标识码: A 文章编号: 1006-8961(2007)12-2031-11

## A Survey of Block-based Motion Estimation

YU Jing, SU Kai-na

(Computer Institute of Beijing University of Technology, Beijing 100022)

**Abstract** Motion estimation refers to estimating 2D motion vector field of the scene or object according to temporal information redundancy in a clipped video. It plays a key role in the super resolution image restoration technique, which maps pixels of all low resolution observations onto corresponding pixels in the reference frame. So its accuracy is highly required. Block-based motion estimation is widely used in super resolution image restoration. This paper introduces the model of block-based motion estimation, summarizes four categories of fast motion estimation methods, describes in detail the search process of several influential algorithms belonging to the first category, and then compares several fast block-matching algorithms.

**Keywords** super resolution image restoration, motion estimation, block-matching algorithm, search pattern

## 1 引言

超分辨率图像复原技术是将变形模糊、有噪、频谱混叠的多幅低分辨率降质图像(或视频序列)融合估计出一幅高质量高分辨率的图像。超分辨率技术在视频监控、卫星图像、视频标准转换、数字镶嵌、医疗数字影像等领域都有着广泛的应用。运动估计就是根据视频序列中时间上相关的信息估计场景或目标的2维运动向量场的过程。运动估计在超分辨

率图像复原中的作用是把所有低分辨率观测帧的像素映射到参考帧的相应位置,实现图像细节的恢复。如果各观测帧表示同一图像细节部分的像素没有映射到参考帧的相同位置,那么就不可能得到正确的恢复结果。因此,精确的运动估计是超分辨率图像复原的关键。运动估计的病态性和复杂性导致运动估计问题成为视频图像处理领域的基本难题之一。运动估计的病态性指可以用许多不同的运动向量场来解释同一幅图像,复杂性指运算量很大。对视频序列进行运动场估计的方法有基于特征的方法、基

基金项目:北京市自然科学基金项目(4031004);北京市教委科技发展计划项目(km200310005006)

收稿日期:2005-11-14;改回日期:2006-03-13

第一作者简介:禹晶(1981~),女,北京工业大学在读硕士研究生。研究方向为计算机应用技术、模式识别与图像处理。

E-mail: yujing@emails.bjut.edu.cn

于光流的方法、基于块的方法、像素递归方法、最大后验概率估计方法等。基于块的方法是超分辨率图像复原中常用的运动估计方法。

## 2 块运动估计模型

运动估计可理解为求同一对象在两帧图像中的位置差,也就是这个对象的“运动”。在基于块的运动估计中,当前帧被划分为图像块,并假设图像块中的各像素具有相同的运动,且只做平移运动。实际上块内各像素的运动不一定相同,也不一定只做平移运动,仅当块较小时,上述假设近似成立。块大小的选择受两个矛盾的约束,块大时,块内各像素做相同平移运动的假设通常不合理;块小时,则易受噪声影响,估计不够可靠,而且运算量增

加。实验结果表明,最合适的块大小为  $8 \times 8$  或  $16 \times 16$ 。块匹配算法 (block-matching algorithm, BMA) 针对当前帧中的每一个图像块,在参考帧的某一区域内搜索最佳匹配块,参考帧中的最佳匹配块相对于当前帧图像块的位移称为当前帧图像块的运动向量 (motion vector, MV)。从计算因素和运动向量的分布考虑,最佳匹配块的搜索不是在整个帧中进行,而是限制在一定的范围内,搜索块的起点一般限于原始图像块起点位置的  $\pm 7$  像素范围内,这个范围称为搜索窗口 (search window)。

块匹配算法的基本思想如图 1 所示,将当前帧划分为图像块,对于当前帧中的每一个图像块,设起始位置的坐标为  $(p, q)$ ,在参考帧的搜索窗口内搜索一个同样大小的最佳匹配块,就可以确定这个图像块的运动向量  $(u, v)$ 。

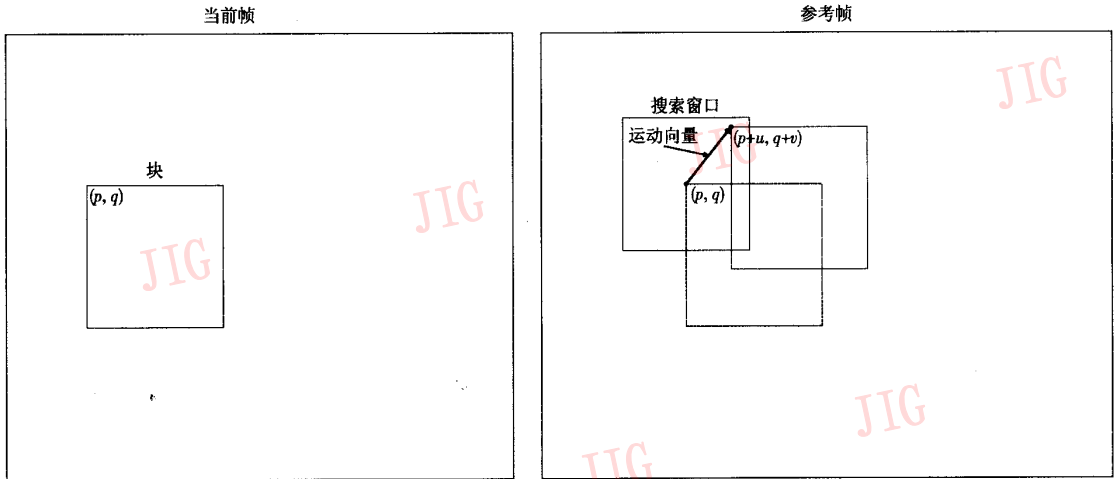


图 1 块匹配算法的基本思想

Fig. 1 Principle of block-matching algorithm

为了寻找最佳匹配块,首先要确定匹配准则,即什么样的块与原图像的块最相似。可以采用的匹配准则有最小均方误差 (MSE) 函数,最小平均绝对值差 (MAD) 函数等。

设当前帧中块的大小为  $N \times N$ ,左上角的坐标为  $(p, q)$ ,运动向量为  $(u, v)$ ,  $I_k(i, j)$  和  $I_{k-\Delta k}(i, j)$  分别为当前帧和参考帧在像素  $(i, j)$  处的灰度值。

MSE 定义为

$$\text{MSE}_{(p,q)}(u, v) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [I_k(p+i, q+j) - I_{k-\Delta k}(p+u+i, q+v+j)]^2 \quad (1)$$

求 MSE 的最小值,使 MSE 达到最小的  $(u, v)$  就

是当前块的运动向量。即

$$[\hat{u}, \hat{v}] = \arg \min_{(u,v)} \text{MSE}_{(p,q)}(u, v) \quad (2)$$

MAD 定义为

$$\text{MAD}_{(p,q)}(u, v) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_k(p+i, q+j) - I_{k-\Delta k}(p+u+i, q+v+j)| \quad (3)$$

估计的运动向量为

$$[\hat{u}, \hat{v}] = \arg \min_{(u,v)} \text{MAD}_{(p,q)}(u, v) \quad (4)$$

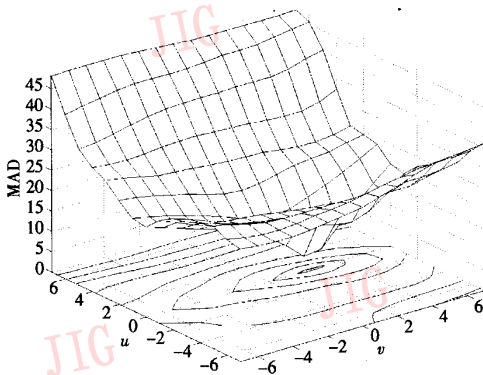
MSE 需要平方运算, MAD 只进行加法和绝对值运算,计算复杂度低,所以在块匹配运动估计中通常采用 MAD 准则计算块匹配误差 (block distortion measure, BDM)。

寻找最佳匹配块最准确的方法是全搜索 (full search) 算法,即对搜索窗口内的所有位置计算块匹配误差,从中找出最小值,其对应的偏移量即为所求的运动向量。若搜索窗口的大小是  $15 \times 15$ ,则需计算 225 个位置处的块匹配误差。采用全搜索算法必然能够找到整个搜索窗口内最佳匹配块的位置,虽然实现起来很简单,但是运算量很大,不适合实际应用,因而各种快速搜索方法迅速产生和发展起来。

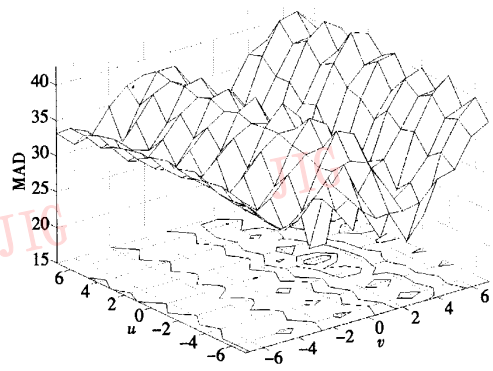
评价块运动估计快速算法通常有 4 个准则:(1) 平均每个像素的 SSE (误差平方和);(2) 每个块的平均搜索点数;(3) 采用快速算法搜索得到的运动向量和全搜索算法搜索得到的最优运动向量相等的块个数占总数的比例;(4) 采用快速算法搜索得到的运动向量和全搜索算法搜索得到的最优运动向量之间的平均距离。根据不同的需求,每种算法力求达到算法复杂度、搜索速率和图像质量的最佳折中。

### 3 块运动估计方法概述

到目前为止,国内外发展了很多块运动估计的



(a) 误差曲面仅有一个局部极小点



(b) 误差曲面有多个局部极小点

图 2 两个块的 MAD (mean absolute difference) 误差曲面

Fig. 2 MAD error surface for two different blocks

这类算法使用固定搜索模式集搜索运动向量,且每个块运动向量的搜索彼此独立。众所周知的搜索算法有 2 维对数搜索 (2D logarithmic search, 2DLOG)<sup>[11]</sup>、3 步搜索 (three step search, 3SS)<sup>[2]</sup>、叉形搜索 (cross search, CS)<sup>[3]</sup>、新 3 步搜索 (new three step search, N3SS)<sup>[4]</sup>、4 步搜索 (four step search, 4SS)<sup>[5]</sup>、梯度下降搜索 (gradient descent search, BBGDS)<sup>[6]</sup>、菱形搜索 (diamond search, DS)<sup>[7,8]</sup>、六

快速算法,与全搜索相比,它们在搜索速率上有很大的提高,而在搜索的准确度上仅有微小的下降。现有的快速算法大致可分为如下 4 类:采用固定搜索模式集的快速块匹配算法,基于块间相关的快速块匹配算法,分级或多分辨率的快速块匹配算法,减少参与块匹配误差计算的像素个数的快速块匹配算法。

#### 3.1 采用固定搜索模式集的快速块匹配算法

这类快速搜索算法是基于误差场单调分布的假设:当搜索位置移向全局极小点的过程中,运动估计的匹配误差单调递减,误差曲面的特性直接影响块运动估计算法的性能。对于图 2(a) 所示的单峰误差曲面,随着搜索位置移向全局极小点,误差函数值单调递减。换句话说,在搜索窗口内有且仅有一个局部极小点。在这种情况下,这类算法经过搜索少量的像素就可以到达全局极小点。但对于图 2(b) 所示的多个局部极小点的误差曲面,这类算法不一定收敛到全局极小点,也就是说搜索很容易陷入某个局部极小点。

边形搜索 (hexagon-based search, HEXBS)<sup>[9]</sup>、十字形菱形搜索 (cross-diamond search, CDS)<sup>[10]</sup>、有效 3 步搜索 (efficient three step search, E3SS)<sup>[11]</sup> 和十字形菱形六边形搜索 (cross-diamond-hexagonal search, CDHS)<sup>[12]</sup>。其中,DS 算法被 MPEG-4VM 采纳。这类算法具有简单性和齐整性的优点,容易实现。

但是,当搜索模式的大小和序列中实际运动的程度不匹配时,就会出现过搜索和欠搜索,影响搜索

的效率和精度。如在 DS 算法中,对于长度小于 2 个像素的运动向量,大菱形搜索模式 (large diamond search pattern, LDSP) 搜索步长太大,引起不必要的搜索过程(过搜索);另一方面,运动较大序列(例如,全局运动)的运动向量靠近中心分布的假设不成立。因此,当搜索这样的序列时,LDSP 的搜索步长太小(欠搜索),导致搜索路径过长。此外,如果它的误差曲面并非单峰(见图 2(b)),LDSP 搜索甚至陷入误差曲面上的局部极小点,导致匹配误差偏大。

### 3.2 基于块间相关的快速块匹配算法

这类算法利用和当前块在空间上和(或)时间上相关的邻近块的运动向量预测当前块的运动向量。通常空间上选择当前块的左、上、右上方的邻近块,时间上选择参考帧中和当前块位置对应的邻近块。通过计算这些邻近块运动向量的统计特征(例如均值、中间值、加权平均值等)<sup>[13,14]</sup>或者根据某准则选择其中一个邻近块的运动向量(例如选择使当前块匹配误差最小的运动向量)<sup>[15~17]</sup>作为当前块运动向量的预测值。获得预测值后,搜索窗口的大小和搜索中心相应地重新定义,在新的搜索窗口内执行缩小的全搜索算法<sup>[15~17]</sup>。这类算法中的一些算法,以预测向量为中心执行缩小的全搜索算法,不重新定义搜索窗口<sup>[13,14]</sup>。

### 3.3 分级或多分辨率的快速块匹配算法

这类方法探索一幅图像不同表示级之间的相关。在分级方法<sup>[18,19]</sup>中,不同级的图像大小相同,块的大小改变,级越低,块越大。这类方法认为在低一级中较大块的运动向量可以预测高一级中较小块的运动向量。但是这种假设经常和实际情况不符,导致匹配性能较低。

多分辨率的方法<sup>[20,21]</sup>利用不同分辨率的图像,块的大小不变。通过对图像下采样,降低图像的分辨率,级越低,图像的分辨率越低。这类方法用低一级所得运动向量作为高一级运动估计的初值,通过由粗到细的过程,可得到原始图像的每个块的运动向量。这类方法有相对较好的运动估计精确度,但在搜索运动向量的细化过程中,需比较缩小的全搜索窗口内所有位置的块匹配误差,因此仍然耗费很大的计算量。均值金子塔<sup>[22]</sup>通过逐级对图像取 $2 \times 2$ 像素块的均值,降低图像的分辨率。由于是通过低通滤波产生的金子塔,因此降低了噪声的干扰。

一种改进的方法结合了运动向量时空相关的方法和多分辨率的方法(也可属于第 2 类)。这种方法的加速比(相对于 FS)达到 150~310, MSE 增加 2%~7%,但这种方法的计算复杂度较高<sup>[23]</sup>。

### 3.4 减少参与块匹配误差计算的像素个数的快速块匹配算法

上述 3 类块运动估计算法都是通过限制搜索位置的个数,降低运动估计的计算量。这类方法通过减少参与匹配误差计算的像素个数加速运动估计。一种方法是在每一个块中分别沿水平和垂直方向按 2:1 的比率下采样,这样计算量将降低为原来的 1/4。这类方法可以和其他类的块运动匹配算法结合达到更高的搜索速率<sup>[24]</sup>。

SEA 算法<sup>[25]</sup>是一种快速全搜索算法,SEA 算法和 FS 算法得到的运动估计结果相同,但它的计算量有较大程度地减小。SEA 算法通过一个有上下界约束的不等式限制了搜索空间,仅需计算参考帧中搜索窗口内满足不等式的搜索位置的块匹配误差,显然,满足不等式的搜索位置的个数比搜索窗口内搜索位置的总数少。SEA 算法不仅降低了计算量,而且能够搜索到全局极小点,得到最佳匹配块对应的运动向量。

## 4 基于误差场单调分布的快速块匹配算法

在国内外提出的块运动估计的快速算法中,其中多数都是基于误差场单调分布假设的。下面按照算法提出的先后顺序,介绍这类方法中几种常用的快速块匹配算法。

### 4.1 3 步搜索算法

3 步搜索(3SS)算法<sup>[2]</sup>是一种非常经典的运动估计快速算法,因为它的简单性和有效性,3 步搜索算法得到广泛使用。它从搜索窗口的原点(0,0)开始,以最大搜索长度的一半为步长,计算中心点及其周围 8 个邻点的块匹配误差,找到最小块匹配误差(minimum block distortion, MBD)点。下一步以该点为中心,步长减半,在缩小的方形上的 9 个点中找 MBD 点,依次类推,直到搜索步长减为 1。若搜索窗口的最大步长为 7,则 3SS 以 4,2,1 为步长序列,经历 3 步完成运动估计,因此称之为 3 步搜索算法。3SS 算法比较的点的总数为 25 个。图 3 是 3SS 搜索 MV 的一个例子。

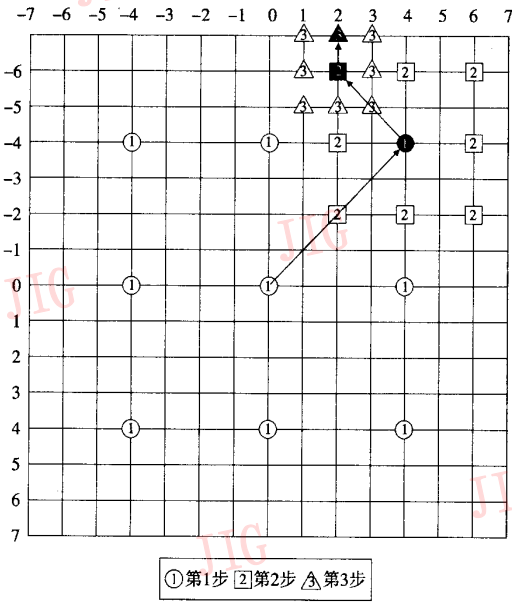
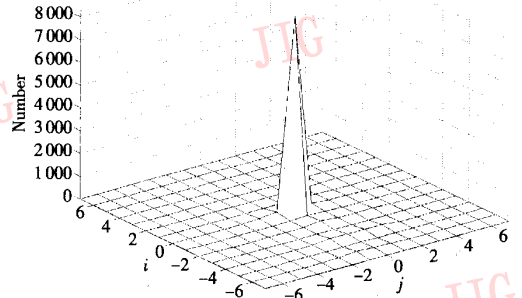
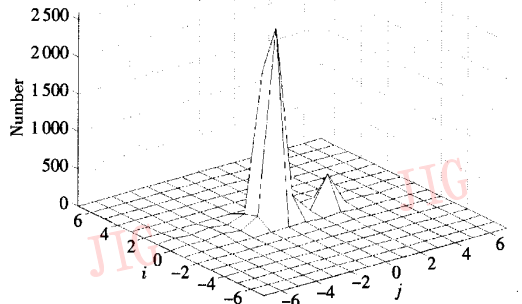


图 3 3SS 算法搜索 MV 的例子  
Fig. 3 A search path of 3SS



(a) Tennis 序列



(b) Caltrain 序列

图 4 Tennis 序列和 Caltrain 序列中 9 帧图像的运动向量分布 (用全搜索)

Fig. 4 The motion vector distribution derived from the full search for 9 frames of Tennis sequence and Caltrain sequence

#### 4.2 新 3 步搜索算法

3SS 算法第 1 步选择的步长较大,对于运动较大的序列,有利于找到全局极小点,但另一方面,对于运动较小的序列,它的搜索可能陷入局部极小点。大量实验结果表明,运动向量的分布具有靠近中心的特性(图 4)。正因为这个分布特性,新 3 步搜索(N3SS)算法<sup>[4]</sup>在 3SS 的第 1 步中加入了以搜索窗口原点为中心的步长为 1 的 8 个邻点参与比较。第 1 步中需比较 17 个点的块匹配误差,若中心是 MBD 点,则算法终止;若新加入的 8 个点之一是 MBD 点,则移动  $3 \times 3$  的方形,使中心落在上一步的 MBD 点上,根据上一步 MBD 点的位置不同,仅需再计算 3 个点或 5 个点的块匹配误差,从中找到 MBD 点,然后终止搜索。否则,搜索步骤与 3SS 相同。虽然在第 1 步中,N3SS 比较了更多的点,但是,当块的运动较小时,第 1 步比较的结果往往就在中间的 9 个点中,后面的比较就大大减少了。

N3SS 通过提前终止搜索加速静止和近似静止块的匹配。对于静止的块,N3SS 需比较 17 个点。对于中心  $5 \times 5$  区域内的运动向量,N3SS 需比较 20 或 22 个点。最坏情况下,N3SS 需比较 33 个点。实验结果表明,N3SS 在性能和速率上都优于 3SS。图 5 是 N3SS 搜索 MV 的一个例子。

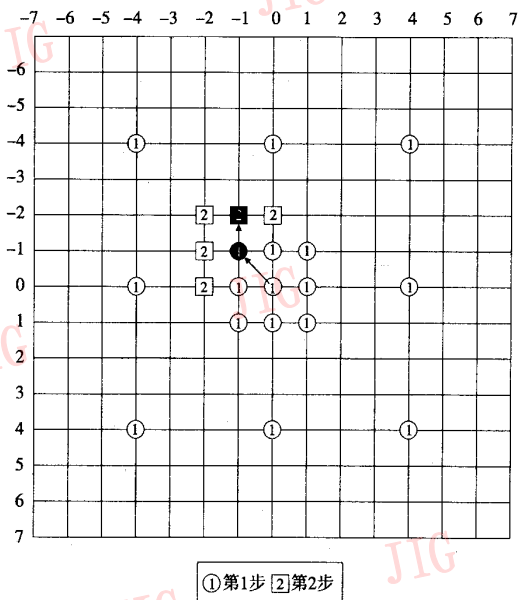


图 5 N3SS 搜索 MV 的例子

Fig. 5 A search path of N3SS

### 4.3 4步搜索算法

4步搜索(4SS)算法<sup>[5]</sup>同样针对3SS算法第1步的搜索步长过大提出了改进。4SS第1步采用和3步搜索一样的方形模式,但步长为2,若MBD点在方形的中心上,或者搜索点到达预定的搜索窗口边界,则步长减为1,最后比较3×3方形上的9个点,从而得到最佳匹配点;否则,以该MBD点作为新的中心继续搜索,步长不变,下一步搜索时只需再计算3个或5个点的匹配误差。

4SS需比较17~27个点,一般情况下比3SS搜索的点少。实验结果表明,4SS的性能优于3SS。4SS算法的实现比较简单。图6是4SS算法搜索MV的一个例子。

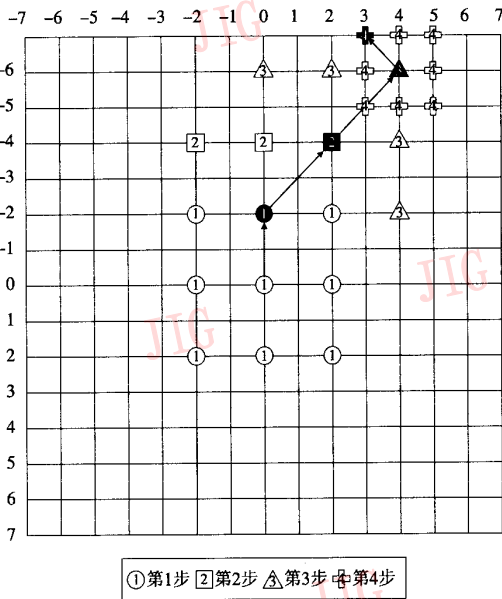


图 6 4SS 算法搜索 MV 的例子  
Fig. 6 A search path of 4SS

### 4.4 梯度下降搜索算法

梯度下降搜索(BBGDS)算法<sup>[6]</sup>完全基于误差场单调分布的假设。BBGDS算法的搜索过程如图7所示,搜索模式是3×3的方形。BBGDS首先将3×3方形的中心放在搜索窗口的原点,计算3×3方形上9个点的匹配误差。若MBD点在方形的中心上,则算法终止;否则,以该点作为新的中心点,继续搜索新形成的3×3方形上的MBD点,直到MBD点在方形的中心上或者搜索点到达预定的搜索窗口边界,每次仅需再计算3个或5个点的匹配误差。

BBGDS算法总是沿着匹配误差最速下降的方

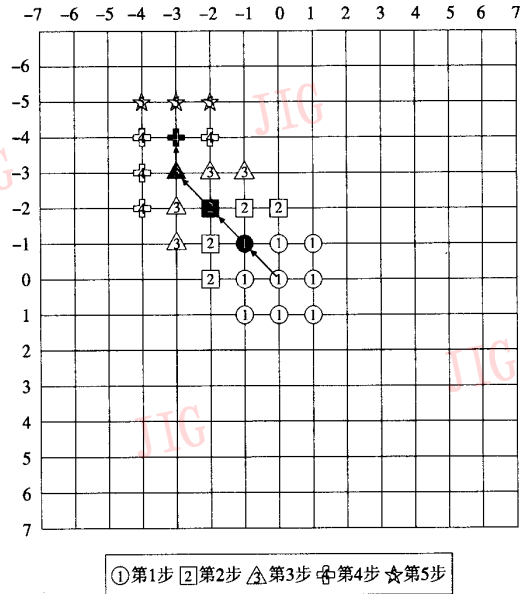


图 7 BBGDS 算法搜索 MV 的例子  
Fig. 7 A search path of BBGDS

向搜索,期望找到全局极小点,因此称之为梯度下降搜索。

### 4.5 菱形搜索算法

菱形搜索(DS)算法<sup>[7,8]</sup>有两种搜索模式,图8(a)和(b)分别称为大菱形搜索模式(LDSP)和小菱形搜索模式(SDSP)。

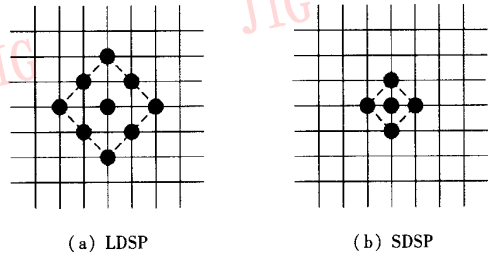


图 8 两种菱形搜索模式  
Fig. 8 Two search patterns used in the DS algorithm

DS算法的搜索过程如下:

- (1) 将LDSP的中心放在搜索窗口的原点,比较LDSP上的9个点的匹配误差。若LDSP中心点的匹配误差最小,则转向第3步,否则,转向第2步。
- (2) 以上一步中的MBD点为中心形成新的LDSP,每次仅需再计算3个或5个点的匹配误差。若LDSP中心的匹配误差最小,则转向第3步,否则,重复执行第2步。

(3) 将 LDSP 切换为 SDSP, SDSP 上 5 个点中的 MBD 点的坐标即为最终求解的运动向量, 指向最佳匹配块。图 9 是 DS 算法搜索 MV 的一个例子。

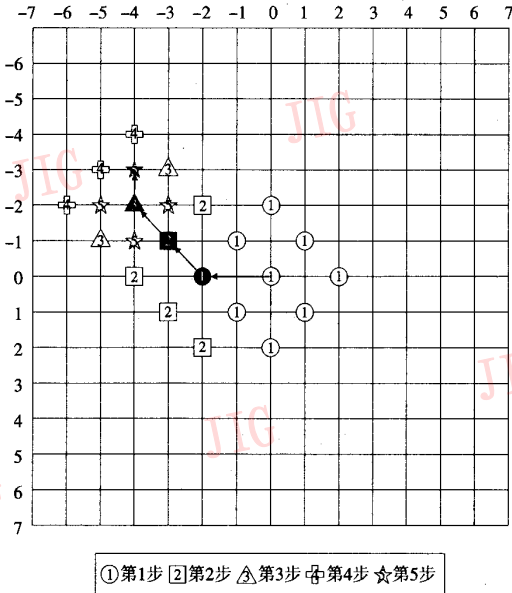


图 9 DS 算法搜索 MV 的例子  
Fig.9 A search path of DS

#### 4.6 六边形搜索算法

近期的研究工作证明了用菱形搜索比前面的方形搜索更加有效。与 BBGDS 算法相比, 菱形搜索在水平垂直方向上具有相对较大的步长, 经过搜索较少的位置可以找到运动较大块的运动向量, 也降低了陷入局部极小点的可能性。但是, 菱形周围的 8 个点 to 中心的距离相差较大, 水平、垂直方向上的步长是 2 个像素, 而对角线方向上的是  $\sqrt{2}$  个像素。在任何方向(水平、垂直、对角)上, 搜索点离全局极小点越近, 匹配误差应越小, 故菱形搜索模式不适合下一步的搜索。文献[9]认为其原因是菱形并不能很好地近似一个圆形, 因此设计了六边形搜索 (HEXBS) 算法<sup>[9]</sup>。

六边形算法有两种搜索模式, 大六边形搜索模式 (LHSP) 和小六边形搜索模式 (SHSP) 分别如图 10(a) 和 (b) 所示, 显然, LHSP 中心周围的 6 个顶点比菱形更近似圆形分布, 此外, LHSP 比 LDSP 少 2 个点。

HEXBS 算法的搜索过程和 DS 算法相同, 用 LHSP 和 SHSP 分别代替菱形搜索中的 LDSP 和 SDSP。图 11 是 HEXBS 算法搜索 MV 的一个例子。

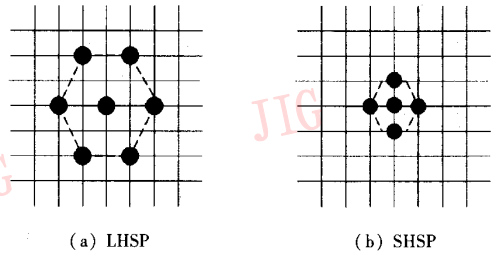


图 10 两种六边形搜索模式

Fig.10 Two search patterns used in the HEXBS algorithm

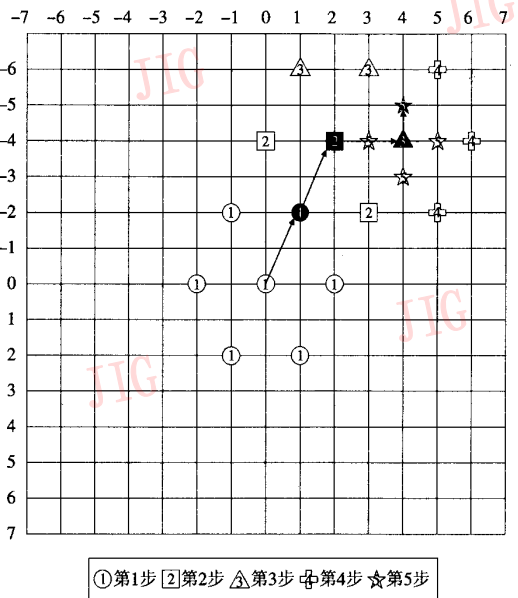


图 11 HEXBS 算法搜索 MV 的例子  
Fig.11 A search path of HEXBS

与 DS 和其他快速搜索算法相比, HEXBS 算法的速率显著提高, 性能相似。

#### 4.7 十字形菱形搜索算法

DS 算法用菱形 (图 12(b)) 搜索运动向量。大量实验结果表明, 中心  $5 \times 5$  的菱形区域内的运动向量中超过 96% 分布在中心十字形区域内, 因此在

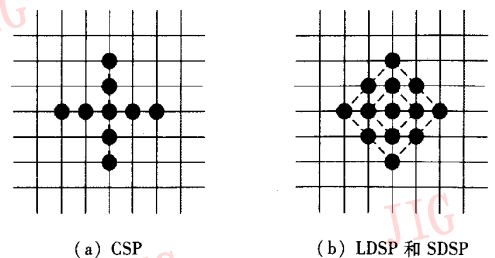


图 12 CDS 算法中的搜索模式

Fig.12 Search patterns used in the CDS algorithm

第 1 步首先采用十字形搜索模式 (CSP) (图 12 (a)), 以提前终止搜索, 减少运动小的块的搜索次数。十字形菱形搜索 (CDS) 算法<sup>[10]</sup> 首先将 CSP 的中心放在搜索窗口的原点, 检查 CSP 上的 9 个搜索点, 若十字形的中心是 MBD 点, 则算法终止, 称为一步终止; 否则, 加入离 CSP 的 MBD 点最近的, 且属于以搜索窗口原点为中心的 LDSP 上的 2 个点 (即坐标为  $(\pm 1, \pm 1)$  的 4 个点中的 2 个点), 参与比较。若上一步的 MBD 点为  $(\pm 1, 0)$  或  $(0, \pm 1)$ , 且此步的 MBD 点和上一步 CSP 上的 MBD 点相同, 则算法终止, 称为两步终止; 否则, 搜索步骤与 DS 算法相同。

对于一步终止和两步终止的运动向量的搜索, CDS 算法分别需比较 9 个和 11 个搜索点。对于没有运动或运动极微小的块的运动向量, 如属于背景区域的块的运动向量, 通过一步搜索就可以得到; 对于运动稍大的块的运动向量, 通过两步搜索也可以得到。图 13 是 CDS 算法搜索 MV 的一个例子。与 DS 算法相比, CDS 算法不仅能保持相似的性能, 有时甚至可以达到更小的误差, 而且提高了搜索速率。

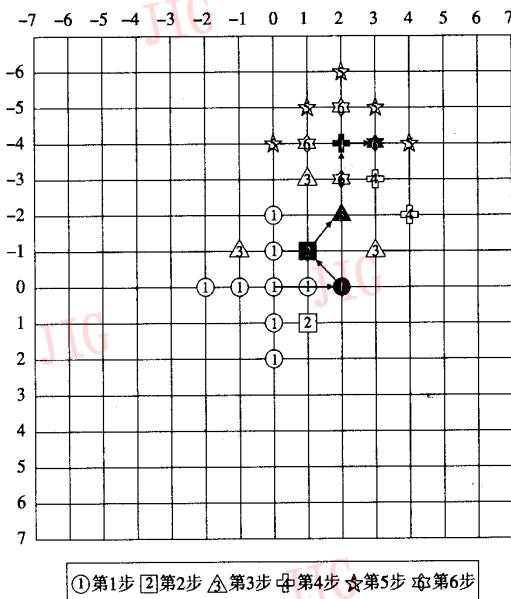


图 13 CDS 算法搜索 MV 的例子  
Fig. 13 A search path of CDS

#### 4.8 有效 3 步搜索算法

因为运动向量的分布具有靠近中心的特性, 有效 3 步搜索 (E3SS) 算法<sup>[11]</sup> 在 3SS 的第 1 步中加入

了 SDSP (图 8 (b)) 上的 4 个点。E3SS 在第 1 步中共比较 13 个搜索位置, 若 SDSP 的中心是 MBD 点, 则算法终止; 若新加入的 4 个点中的某点是 MBD 点, 则将 SDSP 的中心移到上一步的 MBD 点, 如此继续, 直到小菱形的中心是 MBD 点或者搜索点到达预定的搜索窗口边界; 否则, 搜索步骤与 3SS 相同。E3SS 和 N3SS 主要有两点区别: (1) 在靠近中心的区域用小菱形代替方形; (2) E3SS 没有限制小菱形的搜索步数, N3SS 仅移动方形一次。

对于静止的块, E3SS 需比较 13 个点。对于中心  $5 \times 5$  区域内的运动向量, E3SS 需比较 16 ~ 21 个点。最坏情况下, E3SS 需比较 29 个点。实验结果表明, 对于具有较大运动的序列, E3SS 在性能上优于 3SS、4SS、N3SS 和 DS, 对于具有较小运动的序列, E3SS 与 N3SS 和 DS 有相似的性能。

#### 4.9 十字形菱形六边形搜索算法

视频序列中出现较多的是平移 (translation)、缩放 (zooming)、摇摄 (pan)、倾斜 (tilt) 运动。实验结果表明, 若用 DS 算法, 则有大大部分块的运动向量分布在菱形的顶点上, 在菱形搜索模式之前, 首先应用十字形搜索模式 (图 14), 以提前终止搜索, 减少运动小的块的搜索次数, 再用六边形搜索 (图 15), 以减少从菱形顶点开始搜索的额外开销, 因此提出了十字形菱形六边形搜索 (CDHS) 算法<sup>[12]</sup>。

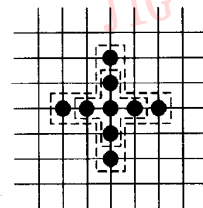


图 14 小十字形搜索模式 (SCSP) 和  
大十字形搜索模式 (LCSP)

Fig. 14 Small cross-shaped pattern (SCSP) and  
large cross-shaped pattern (LCSP)  
used in CDHS algorithms

图 15 提出了两对大六边形搜索模式 (LHSP), 每一对分别包含水平和垂直两个方向。图 15 (a)、(b) 的形式是“扁” (flat) 的, 图 15 (c)、(d) 的形式是“厚” (thick) 的。T-HSP (图 15 (c)、(d)) 以更大的步长搜索, 比 F-HSP 搜索得快, 但准确度降低; F-HSP (图 15 (a)、(b)) 则相反。若用 F-HSP, 则算法称为 CDHS-F; 若用 T-HSP, 则算法称为

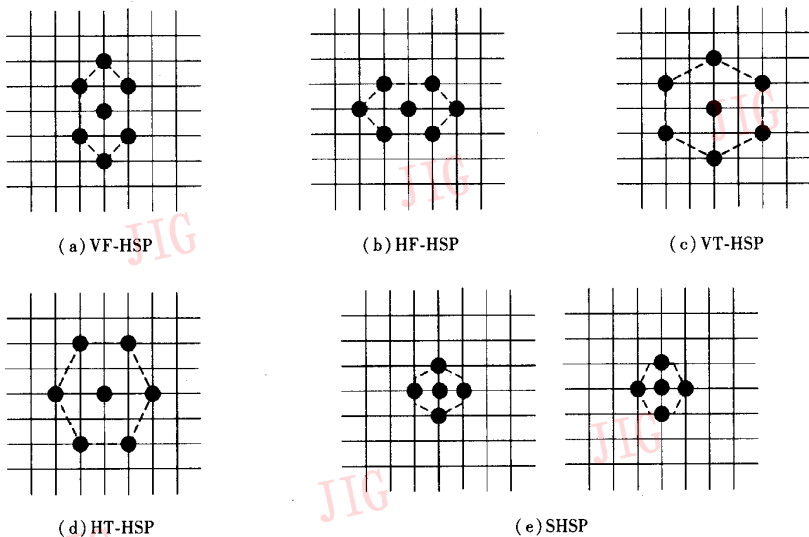


图 15 六边形搜索模式

Fig. 15 HEXBS patterns used in CDHS algorithms

CDHS-T。

CDHS 算法的搜索过程如下:

(1) 开始 将 SCSP 的中心放在搜索窗口的原点,检查 SCSP 上的 5 个搜索点,若中心点的匹配误差最小,则算法终止,称为一步终止。

(2) 大叉形搜索 加入 LCSP 最外面的 4 个点 ( $\pm 2, 0$ ) 和  $(0, \pm 2)$  参与比较,即比较中心在搜索窗口原点的 LCSP 上的 9 个搜索点。

(3) 半菱形搜索 再加入离 LCSP 的 MBD 点最近的,且属于以搜索窗口原点为中心的 LDSP 上的 2 个点参与比较,若上一步的 MBD 点在 SCSP 的端点上,且这一步的 MBD 点和上一步 SCSP 上的 MBD 点相同,则算法终止,称为 3 步终止。

(4) 搜索

第 1 种情况 若上一步用 LDSP 搜索,且菱形边上的点是 MBD 点,则将 LDSP 的中心移到上一步的 MBD 点。

第 2 种情况 若上一步用 LDSP 搜索,且菱形水平(或垂直)方向的顶点是 MBD 点,则将水平(或垂直)方向的 LHSP 的中心移到上一步的 MBD 点。

第 3 种情况 否则,将与上一步相同的 LHSP 的中心移到上一步的 MBD 点。

对于上述的任何一种情况(LDSP→LDSP, LDSP→LHSP, 或 LHSP→LHSP),每次仅出现 3 个新的搜索位置。若 LDSP 或 LHSP 的中心是 MBD 点,则转向第 5 步,否则重复执行第 4 步。

(5) 结束 若上一步用 LDSP 搜索,则切换为 SDSP;若上一步用 LHSP 搜索,则切换为 SHSP。(SDSP 和 SHSP 的形式相同)。这一步 5 个搜索点中的 MBD 点的坐标即为最终求解的运动向量,指向最佳匹配块。

图 16 是 CDHS-T 算法搜索 MV 的一个例子。实验结果表明,CDHS 的搜索速率高于 DS 和 CDS,性能相似。

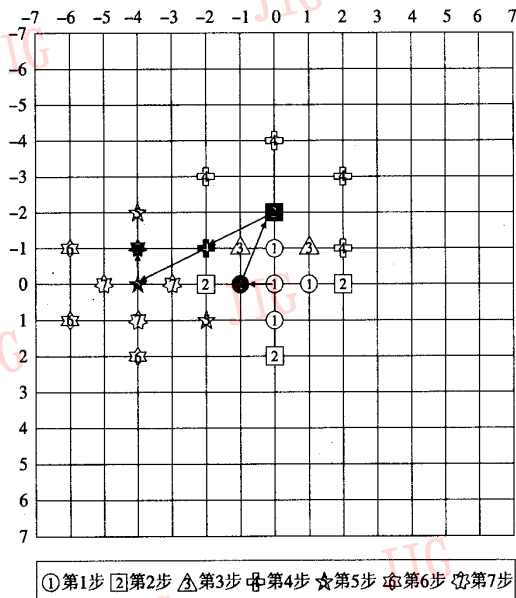


图 16 CDHS-T 算法搜索 MV 的例子

Fig. 16 A search path of CDHS-T

## 5 结 语

超分辨率复原是由一些低分辨率变形图像来融合估计一幅高分辨率的非变形图像,运动估计是超分辨率图像复原中的重要技术,基于块的方法是超分辨率图像复原中常用的运动估计方法。本文描述了块运动估计的模型,介绍了现有的 4 类快速块运动估计的方法,最后详细论述了第 1 类方法中几种主要的基于误差场单调分布的快速块匹配算法。本文将多种快速块匹配算法进行了总结比较,对于大多数视频序列,SEA(快速全搜索)算法的搜索速率相比 FS 算法能提高一倍以上,菱形的搜索精度高于六边形,但搜索速率低于六边形。对于静止或近似静止块较多的视频序列,提前终止搜索的技术不但可以提高搜索速率,而且可以提高搜索运动向量的准确度,如 CDS 和 CDHS 算法。如果当前帧中图像块的运动向量之间的相关性较好,如全局运动的序列,那么第 2 类方法可以同时加速搜索速率和提高搜索精度,但是如果背景中有多个目标,且不同目标的运动各异,那么第 2 类方法的搜索性能反而差。

上述运动估计的结果都是精确到整数像素精度的,超分辨率图像复原需要分数像素精度<sup>[26,27]</sup>的运动估计,这时需要对图像进行  $K$  倍插值,实现精确到  $\frac{1}{K}$  个像素的运动估计,最常用的是半像素的运动估计( $K=2$  的情况)。把低分辨率观测帧上的像素投影到高分辨率参考帧上,这与分数像素精度运动估计的道理是一致的。但是,采用分数像素精度会导致搜索算法的复杂性大大增加。目前,国内外已经提出了许多分数像素精度块运动估计的快速算法,但都不很成熟,在实际应用场合仍然无法获得令人满意的估计结果。块运动估计技术具有相当大的研究潜力,对这类技术的完善和发展是超分辨率复原领域的一个重要研究内容。

### 参考文献 (References)

- 1 Jain J R, Jain A K. Displacement measurement and its application in interframe image coding[J]. IEEE Transactions on Communications, 1981, 29(12): 1799 ~ 1808.
- 2 Koga T, Inuma K, Hirano A, et al. Motion compensated interframe coding for video conferencing[A]. In: Proceedings IEEE National Telecommunication Conference [C], New Orleans, Los Angeles,

- 1981: G5.3.1 ~ G5.3.5.
- 3 Ghanbari M. The cross-search algorithm for motion estimation[J]. IEEE Transactions on Communications, 1990, 38(7): 950 ~ 953.
- 4 Li R, Zeng B, Liou M L. A new three-step search algorithm for block motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1994, 4(8): 438 ~ 442.
- 5 Po L M, Ma W C. A novel four-step search algorithm for fast block motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, 6(6): 313 ~ 317.
- 6 Liu L K, Feig E. A block-based gradient descent search algorithm for block motion estimation in video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, 6(8): 419 ~ 423.
- 7 Tham J Y, Ranganath S, Ranganath M, et al. A novel unrestricted center-biased diamond search algorithm for block motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1998, 8(8): 369 ~ 377.
- 8 Zhu S, Ma K K. A new diamond search algorithm for fast block-matching motion estimation[J]. IEEE Transactions on Image Processing, 2000, 9(2): 287 ~ 290.
- 9 Zhu C, Lin X, Chau L P. Hexagon-based search pattern for fast block motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(5): 349 ~ 355.
- 10 Cheung C H, Po L M. A novel cross-diamond search algorithm for fast block motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(12): 1168 ~ 1177.
- 11 Jing X, Chau L P. An efficient three-step search algorithm for block motion estimation[J]. IEEE Transactions on Multimedia, 2004, 6(6): 435 ~ 438.
- 12 Cheung C H, Po L M. Novel cross-diamond-hexagonal search algorithms for fast block motion estimation[J]. IEEE Transactions on Multimedia, 2005, 7(2): 16 ~ 22.
- 13 Luo L J, Zou C, Gao X Q. A new prediction search algorithm for block motion estimation in video coding[J]. IEEE Transactions on Consumer Electronics, 1997, 43(2): 56 ~ 61.
- 14 Xu J B, Po L M, Cheng C K. Adaptive motion tracking block-matching algorithms for video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1999, 9(10): 1025 ~ 1029.
- 15 Hsieh C H, Lu P C, Shyn J S, et al. Motion estimation algorithm using inter-block correlation[J]. Electronic Letters, 1990, 26(3): 276 ~ 277.
- 16 Tsai J C, Hsieh C H, Weng S K, et al. Block-matching motion estimation using correlation search algorithm[J]. Signal Processing: Image Communication, 1998, 13(2): 119 ~ 133.
- 17 Kim D W, Choi J S, Kim J T. Adaptive motion estimation based on spatio-temporal correlation[J]. Signal Processing: Image Communication, 1998, 13(2): 161 ~ 170.
- 18 Bierling M. Displacement estimation by hierarchical block matching[A]. In: Proceedings of SPIE[C], Hannover, Germany, 1988, 1001: 942 ~ 951.
- 19 Dufaux F, Kunt M. Multigrid block matching motion estimation with

- an adaptive local mesh refinement [A]. In: Proceedings of SPIE [C], Lausanne, Switzerland, 1992, 1818: 97 ~ 109.
- 20 Uz K M, Vetterli M, LeGall D. Interpolative multiresolution coding of advanced television with compatible subchannels [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1991, 1(3): 86 ~ 99.
- 21 Zhang Y Q, Zafar S. Motion-compensated wavelet transform coding for color video compression [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1992, 2(9): 285 ~ 296.
- 22 Nam K M, Kim J S, Park R H, *et al.* A fast hierarchical motion vector estimation algorithm using mean pyramid [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1995, 5(8): 344 ~ 351.
- 23 Chalidabhongse J, Jay Kuo C C. Fast motion vector estimation using multiresolution-spatio-temporal correlations [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(6): 477 ~ 488.
- 24 Liu B, Zaccarin A. New fast algorithms for the estimation of block motion vectors [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1993, 3(4): 148 ~ 157.
- 25 Li W, Salari E. Successive elimination algorithm for motion estimation [J]. IEEE Transactions on Image Processing, 1995, 4(1): 105 ~ 107.
- 26 ISO/IEC 13818-2, Generic Coding of Moving Pictures and Associated Audio[S].