



交流图片




图像和视频编码课程

2014、Pages 255-289

第8章-编码移动图片：运动预测

大卫·R·布尔

显示更多 ∨

 大纲 |  共享  引用<https://doi.org/10.1016/B978-0-12-405906-1.00008-8>

获取权利和内容

摘要

本章展示了如何利用视频信号中的时间相关性通过运动预测提供额外的编码增益。它首先描述了和比较了各种运动模型的优点。基于平移运动模型的块匹配是迄今为止所有标准化视频编解码器中采用的方法，并对其进行了详细研究，其性能的特点是块大小、搜索范围和运动矢量编码的影响。还涵盖了运动估计的替代方法，例如基于相位相关性的方法——用于视频标准转换。接下来，本章从准确性-复杂性权衡的角度比较了降低运动估计复杂性的方法。最后，描述了在传输或存储之前对运动信息进行编码的方式。通篇提供了示例。



关键词

时间相关性; 运动估计; 帧间; 预测; 块匹配; 自相关; 二维对数搜索 (TDL); N步搜索 (NSS); 钻石搜索; 六边形搜索; 运动矢量

在前几章中，我们研究了图像和视频编码的各个重要方面。我们在第4章中看到了如何创建适当的输入格式和颜色空间，我们看到了如何使用变换来装饰和量化图像，例如第5章中的DCT和第6

章中的DWT，在[第7章](#)中，我们描述了如何使用各种形式的熵编码来表示量化输出。本章完成了“图片”，并讨论了我们基本视频编解码器中的最后一个组件——运动预测。运动估计是视频编码的重要组成部分，因为它可以实现[时间装饰](#)。因此，我们不是在每个帧中重复编码非常相似的信息，而是为运动预测后的[剩余信号](#)以及预测参数（运动矢量）编码。

本章首先介绍了时间相关性的概念，并讨论了如何利用它来提供额外的编码增益。然后比较各种运动模型的优点。基于平移运动模型的块匹配是迄今为止所有标准化视频编解码器中采用的方法，并详细研究了其性能，其性能以块大小、搜索范围和[运动矢量](#)编码的影响为特征。还涵盖了运动估计的替代方法，例如基于相位相关性的方法。虽然这些技术很少用于常规视频压缩，但它们值得一提，因为它们用于视频标准转换。

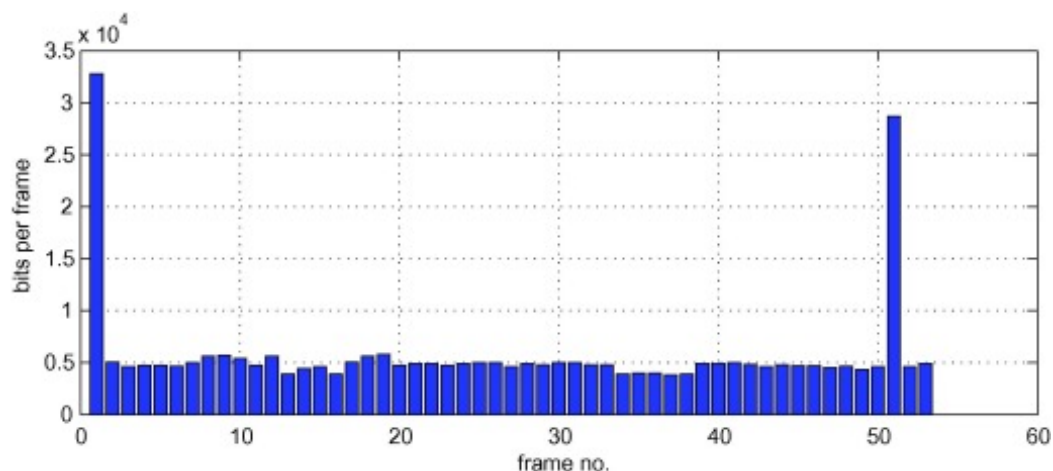
即使在今天，与运动估计相关的主要问题之一是其复杂性——如果不优化，运动估计可以占处理器负载的60-70%。因此，近年来，降低运动估计复杂性的方法受到了高度重视，[第8.4节](#)从准确性-复杂性权衡的角度比较了选定的低复杂性方法。最后，[第8.5节](#)描述了我们如何在传输或存储之前高效编码运动信息。

8.1.时间相关性和利用时间冗余

8.1.1。为什么是运动估计？

对于静止的自然图像，存在显著的空间冗余，我们已经看到这些冗余可以通过变换、[滤波库](#)或其他提供空间[装饰](#)的编码方法来利用。因此，对移动图像序列进行编码的最简单方法是对每帧应用[帧内](#)（静止图像）编码方法（参见Motion JPEG或Motion JPEG2000）。正如我们稍后将看到的，这确实有一些好处，特别是在避免大多数视频编解码器中使用的时间预测循环导致的错误传播方面。当运动模型未能捕捉到场景中的流行运动时，也可以调用它。然而，它通常会导致压缩性能有限。

我们已经[在第1章](#)中看到，对于通过低[带宽通道进行](#)实时视频传输，通常没有足够的容量来独立编码视频序列中的每帧（为了避免闪烁，需要25-30 fps）。解决问题的方法在于利用视频序列中暂时相邻帧之间存在的相关性。这种[帧间](#)冗余可以通过运动预测来进一步提高编码效率。[图8.1](#)显示了这种方法的效果，显示了工头CIF序列的前53帧，该序列编码的GOP长度为50，约为500kb/s。帧1和51被编码为帧内（I帧），其余帧被编码为帧间（或预测帧，也称为P帧）。



[下载：下载全尺寸图像](#)

图8.1。每帧的比特数和帧编码模式的变化。

然而，这种方法出现了一些问题，这些问题将在以下章节中进一步讨论：

- 运动估计产生相关性降低的**剩余信号**。因此，转换编码在**位移帧差**（DFD）信号上的效率低于帧内。
- 运动可以是平移、旋转、变焦和剪切的复杂组合。因此，模型选择是平衡运动预测精度和参数搜索复杂性的关键。
- 照明水平可以从一个帧变化到另一个帧——改变平均**亮度水平**。
- 被遮挡的物体可以变成无遮盖物，可见的物体可以变成遮挡物，混淆运动模型，导致模型失效。
- 运动估计和补偿增加了编解码器的复杂性。
- 运动信息会产生开销，因为它必须与量化变换系数一起传输。

实例8.1 帧内编码和帧间编码对比

考虑一个序列 512×512 以30帧/秒的速度捕获的空间样本，每个彩色通道8位，GOP长度为100帧。如果（对于给定质量），帧内模式下每个像素的平均比特数（包括**运动矢量**开销）为每个彩色通道0.2bpp，在帧间模式下为0.02bpp，则比较此序列的未压缩、压缩和压缩间压缩比特率。

解决方案

1. 未压缩：

$$\text{Uncompressed bit rate} = 512 \times 512 \times 30 \times 24 = 189 \text{ Mbps}$$

2. 帧内压缩：

$$\begin{aligned} \text{Intra-compressed bit rate} &= 512 \times 512 \times 30 \times 0.2 \times 3 \\ &= 4.7 \text{ Mbps} \end{aligned}$$

3. 帧间压缩：

在这种情况下，每100帧中就有1帧是每个彩色通道的帧内@0.2bpp，每100帧中有99帧是每种颜色通道的帧内@0.02bpp。因此：

$$\begin{aligned} \text{Inter-compressed bit rate} &= 512 \times 512 \times 30 \times \left(\frac{0.2 \times 3 \times 1 + 0.02 \times 3 \times 99}{100} \right) \\ &= 514 \text{ kbps} \end{aligned}$$

如果这些数字可以在不影响质量的情况下实现，那么通过利用时间冗余可以实现巨大的编码收益。因此，目标是在为视频序列编码时利用时间和空间冗余。

8.1.2。投影运动和视运动

在现实世界中，物体以三个[空间维度](#)运动，但我们主要感兴趣的是将其运动投影到相机捕获的二维图像平面上。因此，当我们在视频压缩中使用“运动估计”一词时，这通常指物体的投影运动。虽然可以说，以三维跟踪物体可以更准确地描述投影运动，但与这样做相关的模型和方法太复杂，不会引起重大意义。

在视频压缩中，二维运动通常通过观察帧之间信号强度的时空变化来估计。这种方法测量了明显的运动，而不是真正的投影运动，并可能导致歧义。例如，考虑一个围绕其中心旋转的均匀亮度的圆盘。在这种情况下，投影运动显然是非零，而表观运动是零。相反，在图像平面上静止但经历时变照明变化的物体将具有零投影运动，但由于帧之间的强度变化，将显示一些明显的运动。使用表观运动是合理的，因为视频压缩中运动估计的目的不是评估真实运动，而是最小化编码预测剩余所需的比特数。

我们将在下面看到，有许多可能的运动模型可以用于运动估计。然而，最常用的是简单的二维平移模型，其中位移运动矢量， $\mathbf{d} = [d_x, d_y]^T$ 。这通常不是为整个帧计算的，而是为一些较小的块计算的——通常是大小 16×16 像素。所有向量的集合共同表示给定帧的运动场。

8.1.3。了解时间相关性

我们在[第三章](#)中展示了自然视频数据在时间上的相关性，时间[自相关函数](#)通常保持在广泛的滞后值范围内保持高水平。通过运动估计和补偿（MEC），可以利用该属性来补偿相邻帧之间场景中物体的运动，以提高编码效率。例如，[图8.2](#)所示的典型25 fps视频序列（表）中的一对临时相邻帧。尽管这个序列的活动性相对较高——相机从右到左平移，玩家的手、球棒和球都有清晰的运动，但两帧在视觉上看起来非常相似。我们将在以下章节中考虑如何最好地建模此运动，以提高编码增益。



[下载：下载全尺寸图像](#)

图8.2。相邻两个视频帧之间的时间相关性。

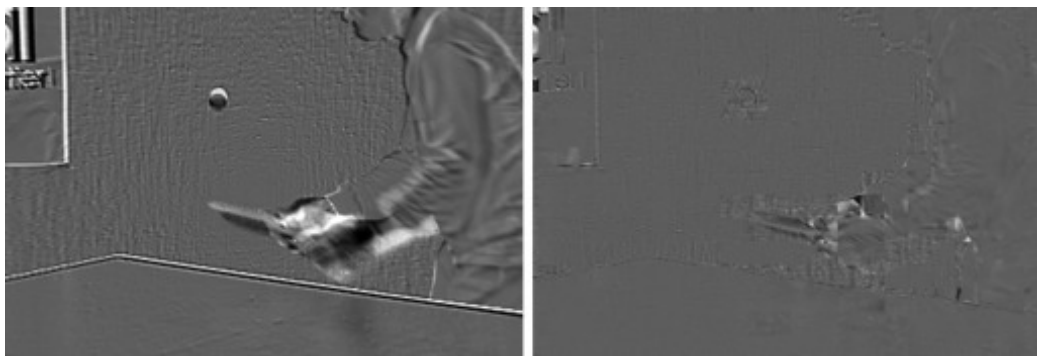
8.1.4。如何形成预测

我们的目标是根据序列中一个或多个其他帧（参考帧）中的值来估计当前编码帧（当前帧）中的像素值。这些通常暂时与当前框架相邻，但不一定。

形成预测的主要方式有两种：

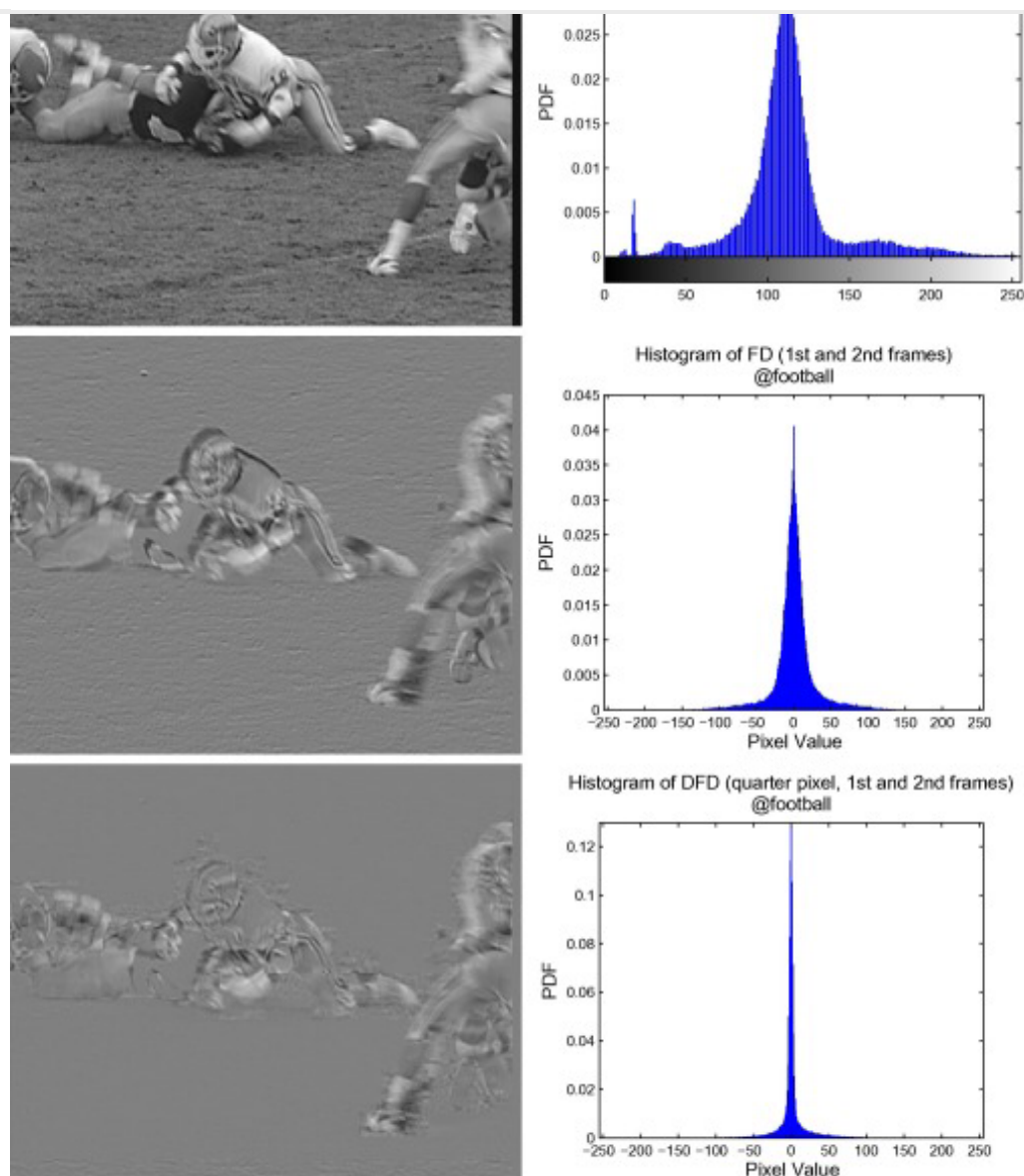
- **帧差异**：这种方法不使用任何运动模型（尽管可以说它可以称为零阶预测）。参考框架不作任何修改，直接用作当前框架的预测。因此，预测误差是当前帧中的像素值与参考帧中同地像素值之间的差值，称为**帧差（FD）**。
- **运动补偿预测**：在这种情况下，假设运动模型，并使用运动估计（ME）来估计参考框架和当前框架之间的运动。一旦估计了运动，就会调用一个称为运动补偿（MC）的过程，根据运动模型使用ME的运动信息来修改参考帧的内容，以产生对当前帧的预测。该预测称为**运动补偿预测（MCP）**或**位移帧（DF）**。预测误差称为**位移帧差（DFD）**信号。

图8.3比较了图8.2所示表中这两种方法的运动残余。可以看出，在运动估计的情况下，**残余能量**显著减少。图8.4强化了这一点，显示了与足球序列的原始帧相比，FD和DFD帧如何修改像素值的pdf。



[下载：下载全尺寸图像](#)

图8.3。帧差异（左）与**运动补偿预测**（右）。



下载: 下载全尺寸图像

图8.4。足球序列中原始帧以及FD和DFD帧的概率分布。

视频压缩中主要使用三种预测类：

- **正向预测：**参考帧在当前帧之前临时出现。
- **反向预测：**参考框架在当前框架之后临时出现。
- **双向预测：**使用两个（或多个）参考框架（前向和后），并以某种方式将候选预测组合起来形成最终预测。

FEEDBACK

在**正向预测**中，假设所使用的运动模型是完美的，并且仅基于平移，则位置上的像素值 $\mathbf{p} = [x, y]^T$ 在当前帧中 k 与上一帧中的**像素位置有关** $k-1$ 作者：²

$$S_k[\mathbf{p}] = S_{k-1}[\mathbf{p} - \mathbf{d}] \quad (8.1)$$

在哪里 \mathbf{d} 运动矢量与位置相对应 \mathbf{p} 在当前帧中。可以生成类似的表达式进行[反向预测](#)。下一节将考虑更复杂的运动模型。

需要注意的是，运动估计问题是假的，不一定有唯一的解——自变量数超过方程数。事实上，解决方案根本不必然存在，例如在未覆盖或遮挡像素的情况下。估计对噪音的存在也很敏感。

8.1.5. 运动估计方法

运动估计的关键问题可以总结如下：

1. **运动模型**：这可以是精确但更复杂的运动模型，如[仿射](#)或[透视](#)，也可以是简单但足够有效的模型，如[翻译模型](#)。在实践中，在大多数情况下都采用平移块匹配方法。一旦选择了运动模型，我们必须决定如何实现它——例如，在[空间或频率域](#)。
2. **匹配标准**：一旦选择模型，我们必须衡量任何候选参数集的匹配度。这可以通过归一化交叉相关性（NCCF）、平方差和（SSD）或绝对差分（SAD）等技术来实现。
3. **支持区域**：这是应用运动模型的像素组，可以从整个帧到单个像素不等。区域可以任意地形成二维或三维，以便最好地反映场景的实际运动特征，但通常以矩形块为基础。在简单的编解码器中，这些通常是 16×16 但可变块大小已变得司空见惯。
4. **搜索范围**：这是一个或多个参考帧中的像素窗口，其中使用匹配标准比较支持区域，以估计运动并参数化模型。搜索范围将取决于所用格式的[时空分辨率](#)和编码内容的活动级别。通常使用矩形区域，最大偏移量在7到63像素之间。
5. **搜索策略**：一旦做出上述所有选择，我们需要执行一种方法，高效地找到最适合运动的模型参数。由于执行[详尽搜索](#)的复杂性，通常使用快速方法。在平移块匹配这些的情况下，包括N步搜索（NSS）、钻石搜索（DS）和[基于六边形的搜索方法](#)（HEXBS）。

在所有情况下，预测准确性和编码开销之间都会存在权衡，必须根据可用的处理能力、视频格式和内容类型做出适当选择。以下各节更详细地探讨了这些问题。

8.2. 运动模型和运动估计

8.2.1. 问题表述

明显的运动可归因于三个主要原因：

1. 全局运动——例如相机运动（平底锅、倾斜、变焦和平移）。
2. 局部运动——场景中物体的局部运动。
3. 照明变化——由于照明、阴影或噪音的变化。

本章描述的模型和方法解决了由于这三个原因导致的组合运动。还有其它方法，最初提取全局运动，然后进行局部细化（例如参考[1]），但这里没有进一步考虑这些。

8.2.2。仿射和高阶模型

图像或图像区域中的表观运动可能很复杂，并经历平移、旋转、剪切和膨胀，以及帧之间的照明变化和其他物体的遮挡（如图8.5所示）。为了表示这种运动特征，有必要使用适当的运动模型。此类模型的例子包括分别由方程（8.2）、（8.3）、（8.4）描述的仿射模型、双线性模型和透视模型。

仿射：

$$\begin{aligned}u &= g_x(x,y) = a_1x + a_2y + a_3 \\v &= g_y(x,y) = a_4x + a_5y + a_6\end{aligned}\tag{8.2}$$

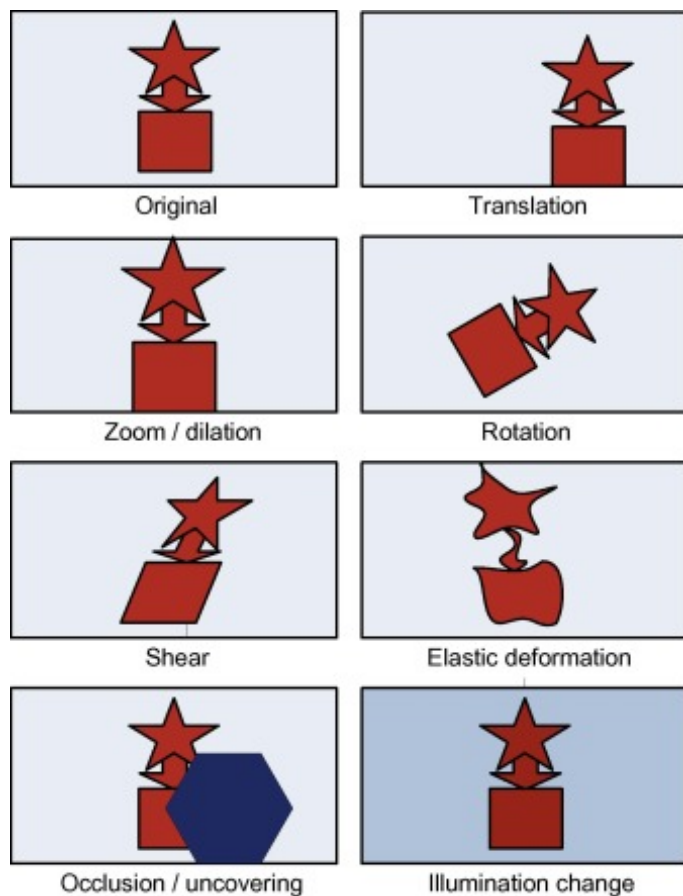
非线性：

$$\begin{aligned}u &= g_x(x,y) = a_1xy + a_2x + a_3y + a_4 \\v &= g_y(x,y) = a_5xy + a_6x + a_7y + a_8\end{aligned}\tag{8.3}$$

视角：

$$\begin{aligned}u &= g_x(x,y) = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} \\v &= g_y(x,y) = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1}\end{aligned}\tag{8.4}$$

位置在哪里 (x,y) 被映射到 (u,v) 使用适当的翘曲参数 $a_1 \cdots a_8$ 。Glasbey和Mardia在Ref中对图像翘曲变换进行了出色的评论。[2]。



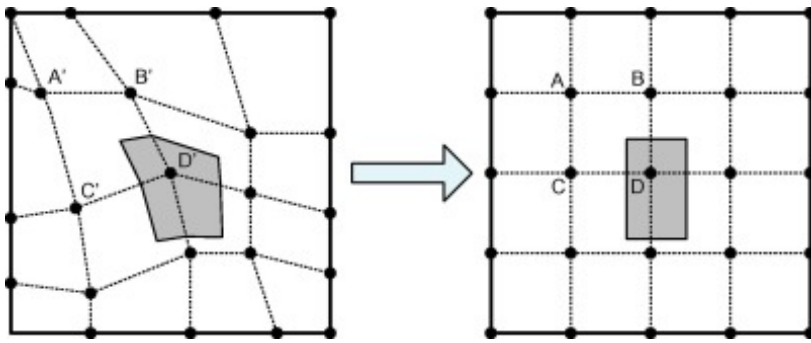
下载：下载全尺寸图像

图8.5。基于运动的图像变形。

基于节点的翘曲

研究文献中广泛报道了高阶模型的使用，但没有一个被采用标准化。然而，最近人们对张和布尔[3]以及维加斯等人[4]以及其他一些人的工作重新产生了兴趣。此外，一些作者提出了使用仿射或八参数模型来建模相机运动的做法，同时使用基于小块的局部运动的平移模型。

在大多数情况下，翘曲参数是通过使用由一组互连的节点点形成的二维可变形网格获得的，如图8.6所示。通过比较当前帧和一个或多个参考帧，使用块匹配运动估计（BMME）等方法来估计每个节点的运动，这反过来又用于查找参数 $\{a_i\}$ 运动模型。然后，该模型用于转换当前块区域中的每个像素。网格可以是连续的或离散的，固定的或自适应的，并且可以采取一系列方法进行节点跟踪。参考文献对这些变化进行了很好的概述和比较。[5]。



下载：下载全尺寸图像

图8.6。基于网格的翘曲。

一旦网格的节点被跟踪，可以通过求解一组同时发生的方程来获得运动参数，例如双线性模型的方程（8.5）中所示。随后，该变换可以使用适当的模型方程应用于补丁中的每个像素，在这种情况下，方程（8.3）。

$$\begin{bmatrix} u_A & u_B & u_C & u_D \\ v_A & v_B & v_C & v_D \end{bmatrix} \quad (8.5)$$
$$= \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \end{bmatrix} \begin{bmatrix} x_A y_A & x_B y_B & x_C y_C & x_D y_D \\ x_A & x_B & x_C & x_D \\ y_A & y_B & y_C & y_D \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

8.2.3. 仅翻译模型

平移模型在实践中被广泛使用，因为它易于参数化，非常适合基于块的常规编码架构，并且重要的是，它很好地适应了大多数场景中的表观运动。后一点值得解释，因为很明显，真实运动、投影运动和明显运动通常不是纯粹的平移运动。

回想一下，我们的主要目的是优化速率-失真性能，这可以通过一个简单的运动模型来实现，其编码开销较低（即参数很少），并且提供低能量预测剩余。平移模型在这方面很成功，因为在典型的视频帧率（25-50 fps）和空间分辨率下，大多数运动都是平移的。这可以在图8.2，图8.3中观察到，图中显示了两个暂时相邻的帧及其剩余图像，经过基于块的平移运动估计。虽然模型失败的地方存在明显的小区域，但在大多数帧中，它工作异常良好，提供非常低的剩余信号。稍后会显示一些进一步的例子，其中这个假设被分解了。

平移模型是透视模型的一个子集，由方程 (8.6) 给出：

$$\begin{aligned}u &= g_x(x,y) = x + a_1 = x + d_x \\v &= g_y(x,y) = y + a_2 = y + d_y\end{aligned}\tag{8.6}$$

还应当指出，包含平移和旋转或平移和膨胀的模型也是可能的。例如，后者由以下方式给出：

$$\begin{aligned}u &= g_x(x,y) = cx + a_1 \\v &= g_y(x,y) = cy + a_2\end{aligned}\tag{8.7}$$

Al-Mualla等人[5]将翘曲方法与平移块匹配进行了比较，两者都基于16 × 16块，用于低比特率编码应用程序。结果汇总于表8.1。比较了三种算法：BMA——一种基本的整数像素块匹配算法；WBA——一种基于翘曲的算法16 × 16使用八参数模型进行节点翘曲的块，BMA-HO-BMA通过半像素精度和重叠的块运动估计增强。可以看出，与基本的BMME算法，翘曲方法确实在预测质量方面有所提高，但这可以通过增强BMME来抵消，例如子像素估计和使用重叠块。就复杂性（处理器执行时间）而言，发现BMA-HO比BMA复杂50%，而WBA比BMA慢50倍。

表8.1。三个测试序列的翘曲和平移模型比较[5]。

PSNR (分B)	秋代	工头	桌子
BMA	39.88	27.81	29.06
WBA	41.45	29.09	29.22
BMA-HO	41.77	29.51	29.87

如上所述，平移运动模型在实际视频压缩系统中应用非常广泛。我们将回到它，并在第8.3节中充分描述它。

8.2.4。像素递归方法

像素递归运动估计方法最早由Netravali和Robbins[6]引入，其基础是使用梯度下降方法迭代最小化预测误差的想法。给定帧（或块）的位移帧差信号：

$$DFD[\mathbf{p},\mathbf{d}] = S_k[\mathbf{p}] - S_{k-1}[\mathbf{p} - \mathbf{d}]\tag{8.8}$$

然后使用最陡峭的下降方法迭代形成每个位置的 \mathbf{d} 运动矢量估计，因此：

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \varepsilon \text{DFD}[\mathbf{p}, \hat{\mathbf{d}}^i] \nabla S_{k-1}[\mathbf{p} - \hat{\mathbf{d}}^i] \quad (8.9)$$

已经观察到，像素递归方法的收敛特性可能与内容（即DFD特性）和步长参数 ε 。它们与光滑强度区域和大位移作斗争；它们还本质上产生密集的运动场，除非使用参数预测方法，否则可能会导致高运动开销。由于这些原因，像素递归方法在实践中没有得到广泛采用。

8.2.5。使用相位相关性（PhC）估计频域运动

原则

相位相关（PhC）的基本概念由库格林和海因斯[7]提出，托马斯为广播应用进行了完善[8]；如图8.7所示。该方法使用二维傅里叶变换来估计相邻帧中块之间的相关性。相关表面的峰值对应于交叉相关滞后，这些滞后显示电流和参考块之间的相关性最高。这种方法的一个优点是，表面可能具有多个峰值，对应于多个候选运动。然后需要某种形式的峰值检测来建立显性运动或运动矢量。PhC方法的优势包括：

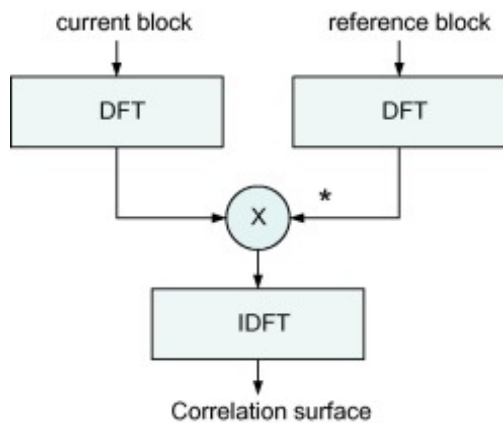
- 它可以检测每个块中的多个运动，方便基于对象的场景分解。
- 可以为每个像素缩小块大小和/或最佳矢量选择。
- 它可以轻松地应对大位移。
- 它为噪声和照明变化提供了鲁棒性，因为傅里叶相位对平均位移不敏感。
- 它通过增加DFT的长度来促进子像素分辨率。

假设完美的平移块运动，对于块或图片， \mathbf{s} ，每个位置相关如下：

$$S_k[x, y] = S_{k-1}[x + d_x, y + d_y] \quad (8.10)$$

在哪里 x 和 y 表示块中的中心像素位置 \mathbf{s} 框架 k ，以及 d_x 和 d_y 是全局翻译帧中块的偏移参数 k 将它与框架中的对齐 $k - 1$ 。来自二维DFT：

$$\mathcal{S}_k(f_x, f_y) = \mathcal{S}_{k-1}(f_x, f_y) e^{j2\pi(d_x f_x + d_y f_y)} \quad (8.11)$$



下载：下载全尺寸图像

图8.7。基于相位相关的运动估计。

在平移运动的情况下，各块中二维傅里叶相位的差异定义了空间频率变量中的平面 f_x, f_y 。运动矢量可以从这个平面的方向来估计，因此：

$$\arg(\mathcal{S}_k(f_x, f_y)) - \arg(\mathcal{S}_{k-1}(f_x, f_y)) = 2\pi(d_x f_x + d_y f_y) \quad (8.12)$$

然而，这需要展开相位，并且不容易识别多个动作。相反，通过二维傅里叶域中的归一化交叉相关函数，使用相位相关方法来估计两个图像块之间的相对移位。在这种情况下，帧之间的相关性 k 和 $k-1$ 由以下定义：

$$C_{k,k-1}[x,y] = S_k[x,y] \star S_{k-1}[-x, -y] \quad (8.13)$$

取两边的傅里叶变换，得到一个复杂的交叉功率谱：

$$\mathcal{C}_{k,k-1}[f_x, f_y] = \mathcal{S}_k(f_x, f_y) \mathcal{S}_{k-1}^*(f_x, f_y) \quad (8.14)$$

按其大小归一化给出了跨功率谱的相位：

$$\angle(\mathcal{C}_{k,k-1}[f_x, f_y]) = \frac{\mathcal{S}_k(f_x, f_y) \mathcal{S}_{k-1}^*(f_x, f_y)}{|\mathcal{S}_k(f_x, f_y) \mathcal{S}_{k-1}^*(f_x, f_y)|} \quad (8.15)$$

假设理想平移运动量 $[d_x, d_y]$ 我们可以从方程 (8.15) 中获得：

$$\angle(\mathcal{C}_{k,k-1}[f_x, f_y]) = e^{2\pi(d_x f_x + d_y f_y)} \quad (8.16)$$

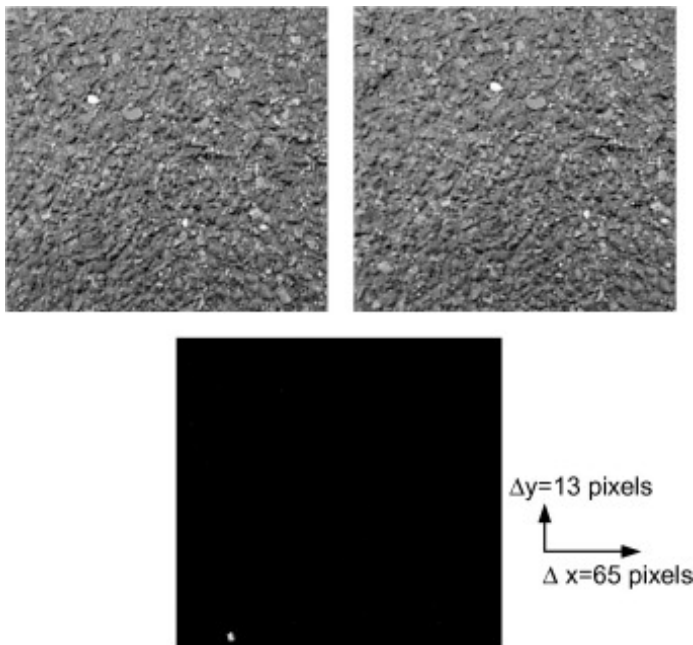
取方程的逆傅里叶变换 (8.16) 得到相相关函数：

$$C_{k,k-1}[x,y] = \delta[x - d_x, y - d_y] \quad (8.17)$$

即相位相关函数产生一个脉冲，其位置代表位移矢量。

应用程序和性能

相位相关性是电视标准转换的首选方法（例如，美国和欧盟帧率之间的帧率转换），主要是因为它产生比BMME更平滑的运动场，熵更低，显著减少了运动伪影。由于使用FFT，PhC可以计算效率高。相位相关性在大规模平移运动的情况下效果良好，可以处理多个运动。然而，使用可变块大小的BM现在被认为是一种更灵活的方法，在现代视频编码标准中，它更适合常规运动估计。PhC的一个例子如图8.8所示。



[下载：下载全尺寸图像](#)

图8.8。基于相位相关运动估计的例子。顶部：用大约翻译相机拍摄的柏油碎路面的输入帧。底部：相相关表面。

8.3. 块匹配运动估计 (BMME)

8.3.1. 平移块匹配

运动矢量方向

图8.9说明了我们将采用前运动矢量定向的约定。运动矢量表示参考系中最近匹配块相对于当前块位置的位移（反转真实运动的极性）。我们将假设 $[0,0]$ 矢量对应于搜索网格的中心，并在 x 和 y 分别是正的和正的。

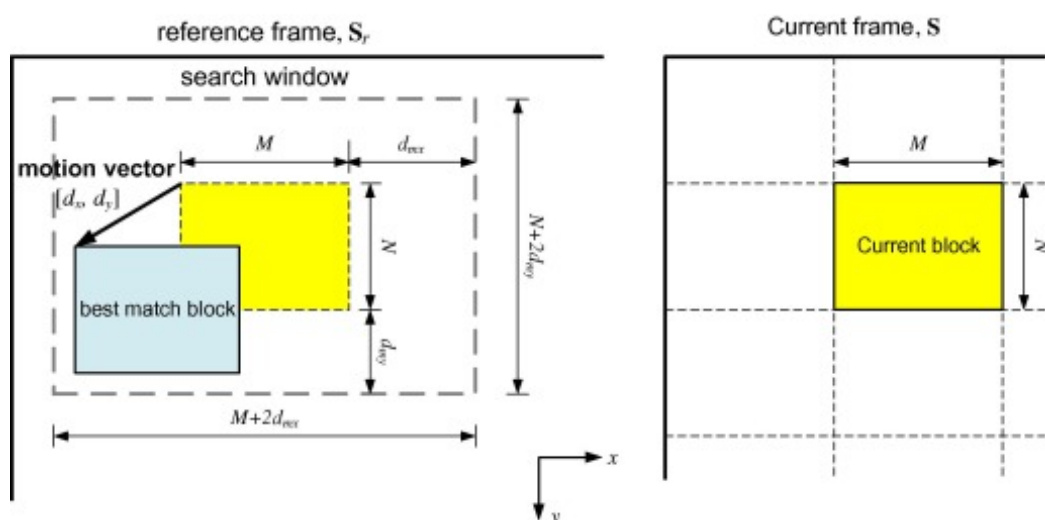


图8.9。平移块匹配运动估计（BMME）。

支持区域——搜索窗口的大小

直观地，我们希望搜索整个参考图片，以找到与当前块的最佳匹配。然而，在大多数情况下，这既不切实际，也没有必要。考虑一个1080p25序列，一个对象以1s的方式穿过屏幕；使用1920个水平像素，这意味着该对象水平移动 $1920/25 = 77$ 每帧像素。因此，人们可能会争辩说，这种运动将决定一个搜索窗口 ± 77 像素来预测它。在实践中，许多其他因素也起作用了。首先，以1秒的速度穿过屏幕的物体非常快，很少发生（至少不会出现在定向内容中）；其次，如果确实发生，将有显著的[运动模糊](#)，因此无论如何，该物体永远无法忠实地描绘；第三，运动掩蔽效果将存在，这进一步减少了忠实复制的必要性。因此，通常使用折衷窗口，支持区域满足大多数运动场景，同时限制搜索复杂性。

如图8.9所示，搜索网格通常限于 $[\pm d_{mx}, \pm d_{my}]$ 在大多数情况下， $d_{mx} = d_{my} = d_m$ 。价值 d_m 取决于图像分辨率和活动；例如，在广播应用程序中， d_m 通常从低运动的15到高（运动）运动的63不等。

8.3.2。匹配标准

块失真度量（BDM）

对于块匹配运动估计，通常假设均匀运动和平移运动模型。我们需要为当前块选择最佳的运动矢量，这将对应于最低的[残差失真](#)。这用某种形式的块失真度（BDM）来评估，该测量公式如下：

$$\text{BDM}(i, j) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} g(s(x, y) - s_r(x + i, y + j)) \quad (8.18)$$

在哪里 $g(\cdot)$ 是用于计算BDM的函数， i 和 j 是当前运动矢量候选的偏移量，以及 x, y 是当前块内的局部坐标。

我们选择最小化当前块BDM的运动矢量：

$$\mathbf{d} = [d_x, d_y]^T = \arg \left(\min_{\forall(i, j)} (\text{BDM}(i, j)) \right) \quad (8.19)$$

绝对差值与平方差值测量

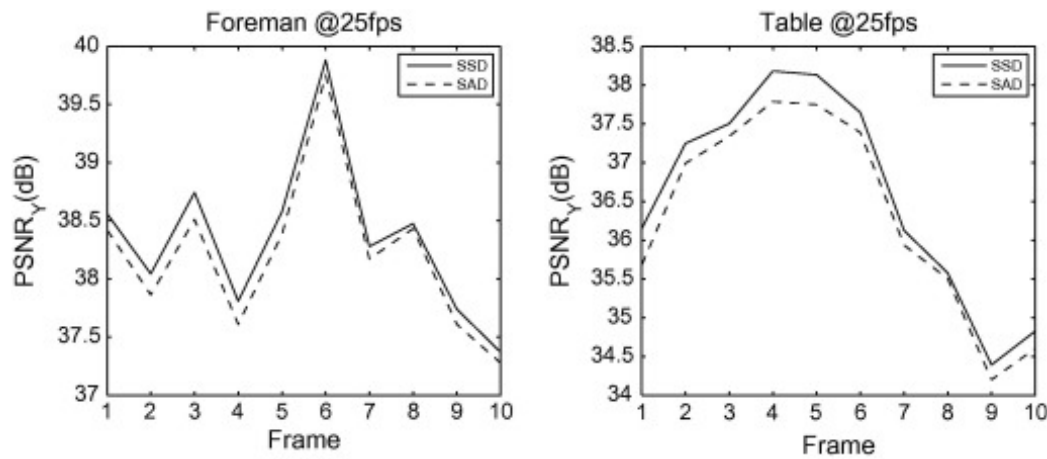
有许多可能的块失真函数可用于评估运动估计后的误差残差。这些包括归一化交叉相关性（NCCF）、[平均平方误差](#)（或平方差分之和（SSD））和绝对差分之和（SAD）。BDM函数的选择是关键，因为它影响搜索复杂性和预测准确性。

最常见的两个函数是SAD（方程 [\(8.20\)](#)）和SSD（方程 [\(8.21\)](#)）。

$$\text{SAD}(i,j) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |s(x,y) - s_r(x+i,y+j)| \quad (8.20)$$

$$\text{SSD}(i,j) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (s(x,y) - s_r(x+i,y+j))^2 \quad (8.21)$$

图8.10说明了这两个函数在两个不同序列的预测精度上的差异。可以观察到，尽管SSD指标在质量方面一直优于SAD，但实际上这两种方法之间几乎没有区别。此外，SAD指标只需要 $3NM$ 每个块的基本操作（添加、子、Abs）进行计算，而SSD需要 NM 乘法加 $NM-1$ 添加。因此，经常选择SAD函数而不是SSD。



[下载：下载全尺寸图像](#)

图8.10。匹配功能的效果：SAD vs SSD。

8.3.3。全搜索算法

最简单的BMME算法是基于全搜索或全搜索。在这种情况下，通过彻底检查所有可能的问题，可以找到误差表面的全局最小值 $(2d_m + 1)^2$ 搜索网格中的运动矢量候选者。

8.3.4。块运动场和误差面的性质

块运动场

对于自然场景，块运动场一般是平滑的，并且变化缓慢。这是因为相邻块的运动矢量之间通常具有很高的相关性。图8.11以25帧/秒捕获的工头序列为例说明了这一点。

$\rho_X = 0.56$ $\rho_Y = 0.33$	$\rho_X = 0.69$ $\rho_Y = 0.49$	$\rho_X = 0.64$ $\rho_Y = 0.46$
$\rho_X = 0.66$ $\rho_Y = 0.48$	Current Block	$\rho_X = 0.72$ $\rho_Y = 0.63$
$\rho_X = 0.64$ $\rho_Y = 0.43$	$\rho_X = 0.76$ $\rho_Y = 0.61$	$\rho_X = 0.68$ $\rho_Y = 0.56$

[下载](#): [下载全尺寸图像](#)

图8.11。典型块运动场（工头为25 fps）的自相关值。

改编自参考文献。[5]。

块大小的影响

图8.12显示了不同块大小（ 8×8 vs 16×16 ）在预测质量和运动开销方面。可以看出，较小的块大小通常会导致更好的预测精度，但只会以大幅提高比特率为代价。预测精度的提高是由于较小的块可以更好地处理多个运动、复杂运动（即提供更适合模型）以及纹理和非纹理区域的不同属性。由于速率和失真之间的冲突，许多现代压缩标准使用可变块大小进行运动估计，以及速率失真优化（RDO）技术，这些技术选择优化速率失真权衡的块大小。第九章和第十章进一步讨论了这一点。

示例8.2 全搜索运动估计

给定以下几点 2×2 当前帧S中的块，以及相应的搜索窗口W，确定此块的运动矢量d。使用带有SAD错误度量的完整搜索策略。

$$\mathbf{W} = \begin{bmatrix} 1 & 5 & 4 & 9 \\ 6 & 1 & 3 & 8 \\ 5 & 7 & 1 & 3 \\ 2 & 4 & 1 & 7 \end{bmatrix}; \quad \mathbf{S} = \begin{bmatrix} 3 & 9 \\ 1 & 4 \end{bmatrix}$$

解决方案

例如，考虑候选运动矢量 $\mathbf{d} = [0,0]$ 。在这种情况下，位移帧差信号由以下方式给出：

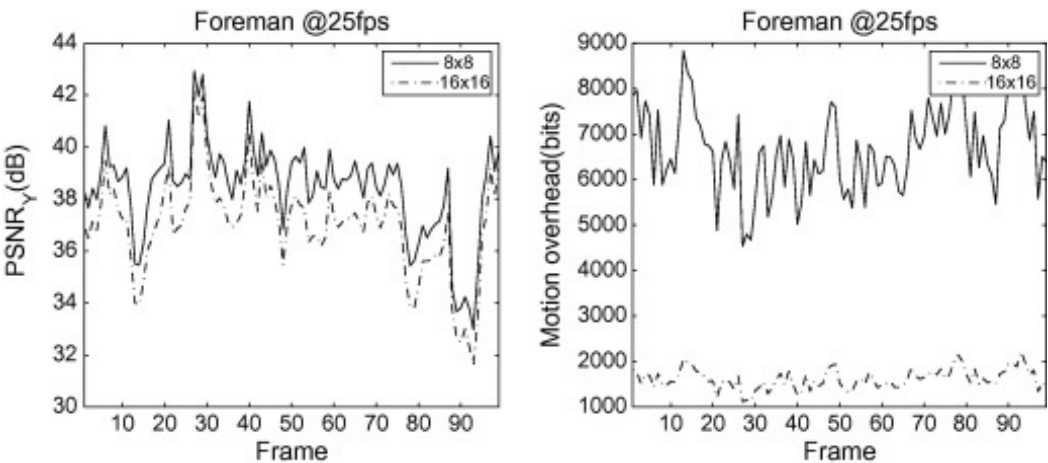
$$\mathbf{E} = \begin{bmatrix} 3 & 9 \\ 1 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 3 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ -6 & 3 \end{bmatrix}$$

和 $\text{SAD} = 17$ 。所有其他类似 $[d_x, d_y]$ 偏移量：

i	j	$SAD(i,j)$	i	j	$SAD(i,j)$
-1	-1	14	0	1	18
-1	0	18	1	-1	7
-1	1	5	1	0	2
0	-1	8	1	1	11
0	0	17			

[下载：下载全尺寸图像](#)

因此，这个块的运动矢量是 $\mathbf{d} = [1,0]$ 。



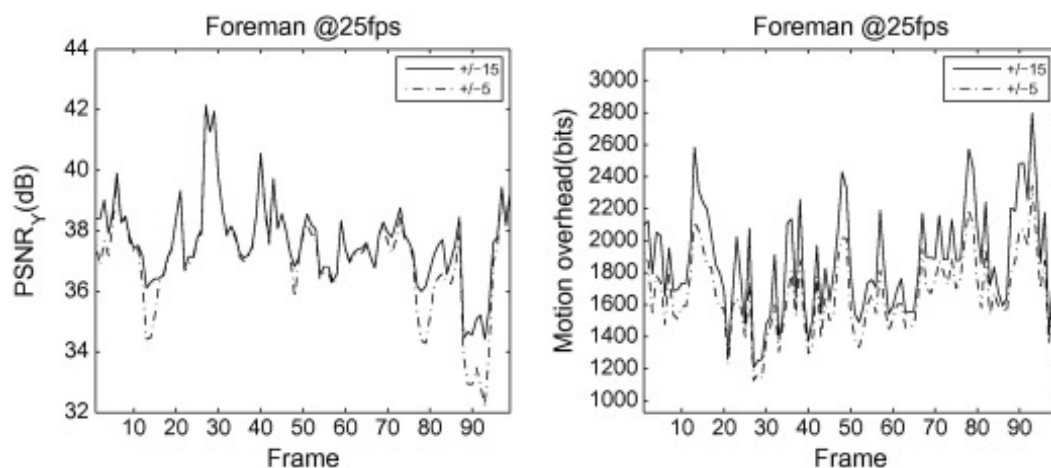
[下载：下载全尺寸图像](#)

图8.12。块大小对运动估计性能的影响。

搜索范围的效果

搜索网格大小由当前块位置的最大运动矢量偏移量决定，记为 d_m 。价值 d_m 也影响复杂性和准确性，因为完全搜索的复杂性会随着它的平方而增加（有 $(2d_m + 1)^2$ 待评估的候选人）和大多数快速搜索方法都会线性增长。少量的 d_m 会表示在搜索寡妇中不会捕获快速移动的对象，导致预测质量差，运动失效。相比之下，一个很大的价值 d_m 可能效率低下，导致计算复杂度过高。

搜索窗口大小的选择取决于内容类型和视频格式。例如，体育内容通常需要比脱口秀更大的窗口，对于相同的帧率，HDTV将需要比SDTV更大的窗口。图8.13显示了CIF工头序列的搜索窗口大小以25帧/秒的效果。可以看到，一个小 ± 5 在这种情况下，窗口在大多数情况下都令人满意，除了少数时间区域（大约12、50、80、90帧）外，在这些区域，较高的运动会导致质量显著下降。较大的搜索窗口将处理这些快速运动，但确实需要少量增加比特率，以处理平均运动矢量大小的增加。

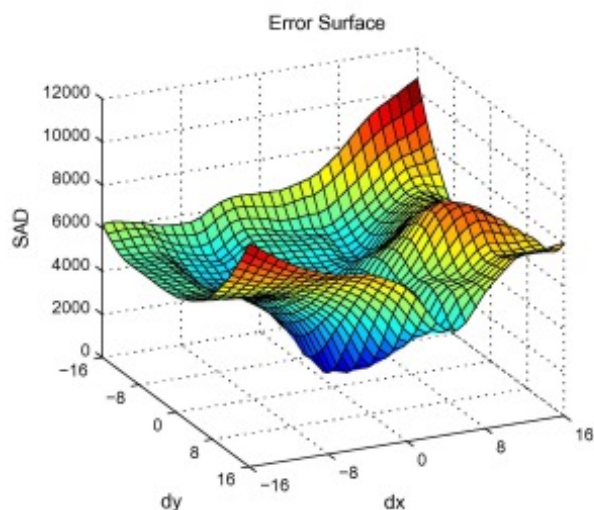
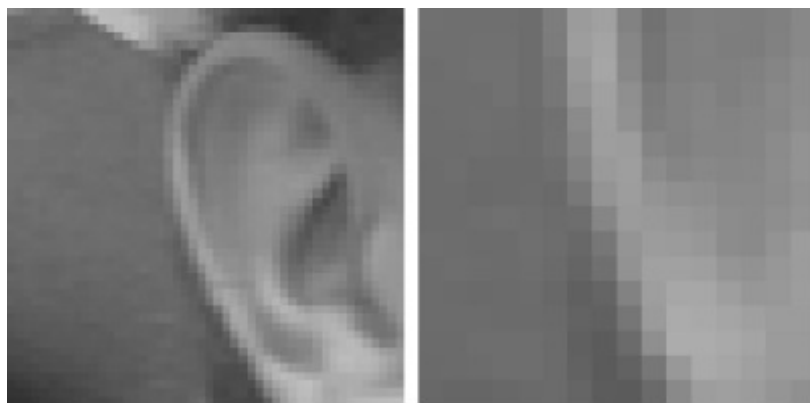


[下载: 下载全尺寸图像](#)

图8.13。搜索范围对预测精度和编码比特率的影响

运动残差误差面

每个块的错误表面（即搜索网格中每个[候选向量](#)的BDM数字）通常是多模态的，包含一个或多个本地最小值。这种情况可能是由于局部纹理的周期性，块内存在多个运动，或者只是运动模型和表观运动之间的不匹配。此外，全局最小值和最大 SAD 值可能差异很大，甚至在单个帧中处于序列中。错误表面的例子如[图8.14](#)所示，以及相应的当前块（右上角）和搜索窗口（左上角）。我们可以观察到，在所示的例子中，SAD表面确实是多模态的，值从1000以下延伸到11000以上。

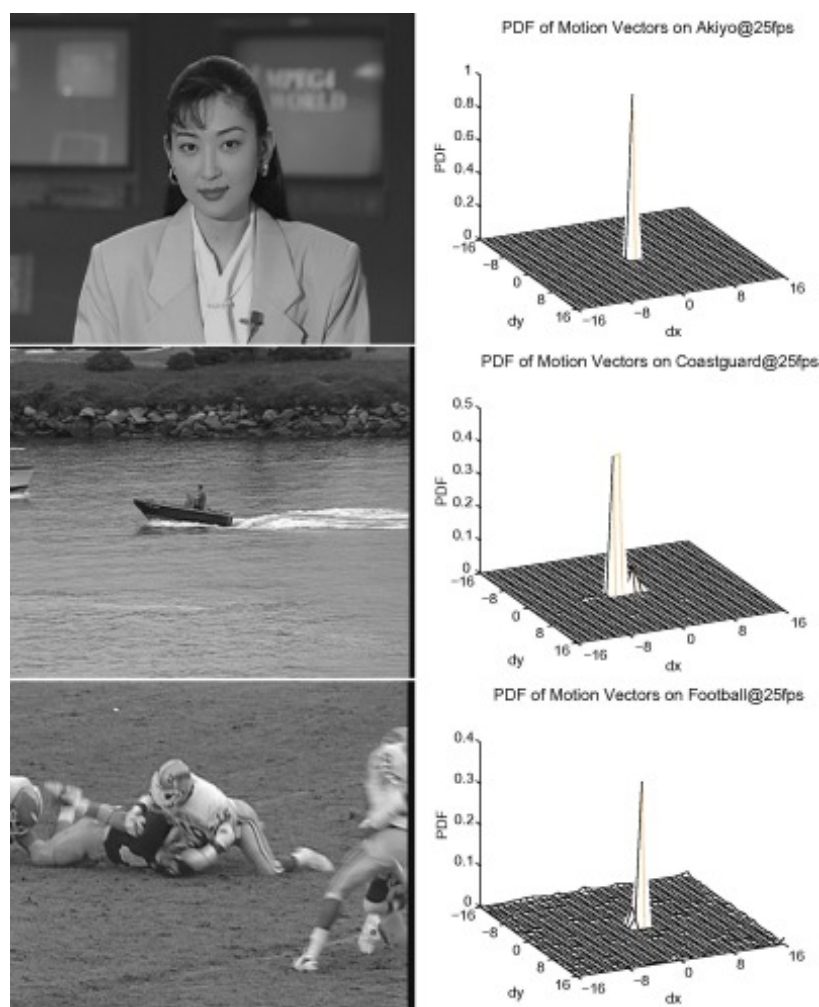


[下载：下载全尺寸图像](#)

图8.14。典型的运动残差误差面 16×16 工头序列的块（CIF为25 fps， $\pm 16 \times \pm 16$ 搜索窗口）。左上角：搜索窗口。右上角：当前块。

运动矢量概率

块运动场的分布是中心偏置的，这意味着较小的位移比较大的位移更可能， $[0,0]$ 矢量（在没有全局运动的情况下）最有可能。对于以25 fps编码的一系列序列，此属性如图8.15所示。pdf中的峰值 $[0,0]$ 对于秋代和足球序列来说是显而易见的，而由于海岸警卫队的相机平底锅，有明显的偏移量。低活动序列秋代在pdf中显示一个很大的峰值，在 $[0,0]$ ，其他地方的值非常小。另一方面，足球序列显示，由于相机平移，峰值和清晰的元素要低得多，以及由于场景的复杂性，距离 $[0,0]$ 的显著pdf能量。海岸警卫队序列显示的主峰更宽，由于相机平底锅而偏置，但能量也远离 $[0,0]$ ，这主要是由于难以估计动态水纹理的随机运动。

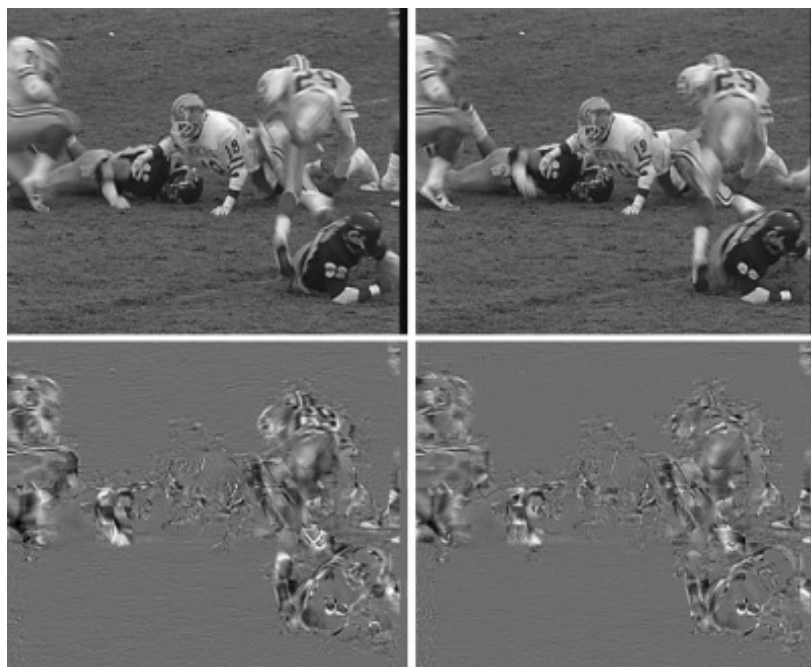


[下载](#): [下载全尺寸图像](#)

图8.15。秋水、海岸警卫队和足球序列的[运动矢量pdf](#)。

8.3.5。运动失败

尽管基于块的平移运动模型在大多数情况下工作良好，并且可以提供出色的速率-失真性能，但在某些情况下，运动特性严重失败。这种情况如[图8.16](#)所示。这两帧来自*Football*，捕捉了此类内容中典型的高活动，DFD即使在子像素运动估计后，也显示了重要的[伪影](#)。这些工件将导致DFD具有非常低的[自相关系数](#)，因此装饰变换（DCT或其他）将表现不佳。



[下载：下载全尺寸图像](#)

图8.16。足球序列运动模型失败示例。左上角：当前帧，右角：参考帧。左下角：全像素估计，右角：半像素估计。

那么，我们如何应对这种情况呢？基本上，两种方法可以单独使用或组合使用。第一个比较了帧内和帧间编码的速率失真性能，并选择了最佳编码模式。第二个使用循环解块过滤器来删除参考帧中引入的边缘伪影。这些方法在第9章和第10, [章第9章第10 and 章](#)进一步讨论。

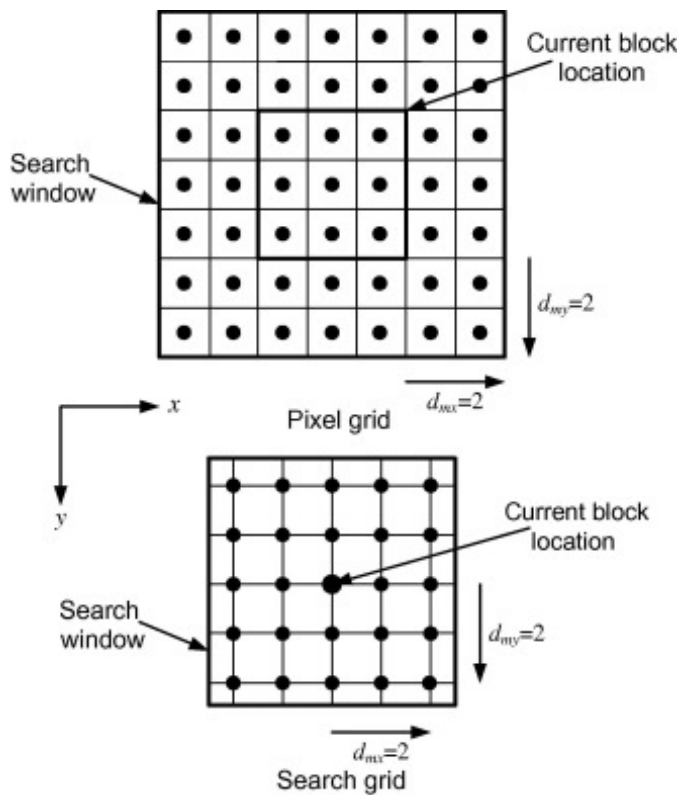
8.3.6。受限和不受限制的向量

如果块位于帧边缘或靠近帧，部分搜索区域将超出帧边界。在标准运动估计中，矢量仅限于帧内。然而，通过边缘像素外推来人为地扩展搜索窗口，从而支持使用不受限制的运动矢量，可以获得好处，特别是在相机运动的情况下。外部像素通常基于边界值的零阶或对称外推。

8.4。降低复杂性运动估计

8.4.1。像素网格和搜索网格

在我们研究高级搜索技术之前，值得花一点时间区分像素网格和[搜索网格](#)。两者都代表了运动搜索的支持区域，但我们将使用后者作为描述从现在开始搜索方法操作的基础。像素网格只是图像矩阵，其中每个元素表示搜索窗口中的像素值。相比之下，搜索网格以其偏移量表示搜索窗口。在不失去通用性的前提下，假设一个方形的搜索窗口，搜索网格的尺寸是 $2d_m + 1$ 通过 $2d_m + 1$ 。这对应于 N 通过 N 块大小，到带有尺寸的搜索窗口 $2d_m + N$ 通过 $2d_m + N$ 。这两个表示如[图8.17](#)所示。



下载：下载全尺寸图像

图8.17。像素网格和搜索网格。

8.4.2。全面搜索的复杂性

对于每个运动矢量，都有 $(2d_m + 1)^2$ 搜索网格中的运动矢量候选者。在每个搜索位置 MN 像素进行比较，每次比较需要三个操作（Sub、Abs、Add）。因此，每个宏块的总复杂性（以基本算术运算计算）是 $3MN(2d_m + 1)^2$ 。

因此，对于图片分辨率 $I \times J$ 和帧率 F fps，总体复杂性（在每秒操作中）由方程 (8.22) 给出：

$$C_{FS} = 3IJF(2d_m + 1)^2 \quad (8.22)$$

示例8.3 完全搜索复杂度

考虑一个具有以下规格的视频序列： $M = N = 16, I = 720, J = 480, F = 30, d_m = 15$ 。计算该序列的全搜索运动估计算法的复杂性。

解决方案

如上所述，完整的搜索复杂性由 $3IJF(2d_m + 1)^2$ 每秒操作。将给定序列的值替换为处理器加载 29.89×10^9 每秒操作！

8.4.3。降低搜索复杂性

正如我们不久将看到的，搜索复杂性可以大大降低，但通常以牺牲次优性能为代价。存在许多方法，这些方法可以基于：

1. **降低复杂性 BDM**：在实践中，很难改进 SAD 的使用，SAD 已经在复杂性和性能之间提供了很好的权衡。这一类别中一个有趣的方法是陈和萧[9]，其中BDM评估中使用的像素数量因块活动而异。
2. **子采样块运动场**：这种方法利用块运动场的平滑性。在选择最小BDM之前，只评估了一小部分块候选者，并插入其余部分。例如，刘和扎卡林[10]使用棋盘模式来评估一半块的运动矢量。这将搜索复杂度降低了2倍，预测质量几乎没有下降。
3. **分层搜索技术**：这类方法使用多分辨率方法，在更精细的尺度上逐步细化运动矢量估计。Nam等人介绍了使用平均金字塔的此类方法示例。[11]。据报道，这些方法还提供了更均匀的块运动场，更好地表示真实运动。
4. **一组简化的运动矢量候选**：此方法限制使用某种形式的有向搜索模式计算的块数。这一类别中大多数快速的ME算法都基于平滑和单模态误差表面的假设（即误差表面随距离最小BDM点的距离而单调增加）。我们已经看到，这个假设并不总是有效的，算法可能会被困在局部极小值中（见图8.14），因此使用的搜索模式必须考虑到这一点。这种类型的最早方法是??那教和??那教提出的二维对数（TDL）搜索[12]，随后出现了许多变体。这类方法被广泛认为在复杂性和预测质量之间提供了很好的折衷方案，并在实践中被广泛使用。以下小节介绍了该类别的选定方法。
5. **智能初始化和终止**：再次，这些利用了块运动场的属性。智能初始化标准用于在预测运动矢量的位置而不是搜索网格中心开始搜索。此类方法可以与（4）一起使用，以帮助避免局部最小值。

读者被引导到参考文献[5]，以详细比较这些方法。在以下章节中，我们将更详细地研究一些更流行的技术。

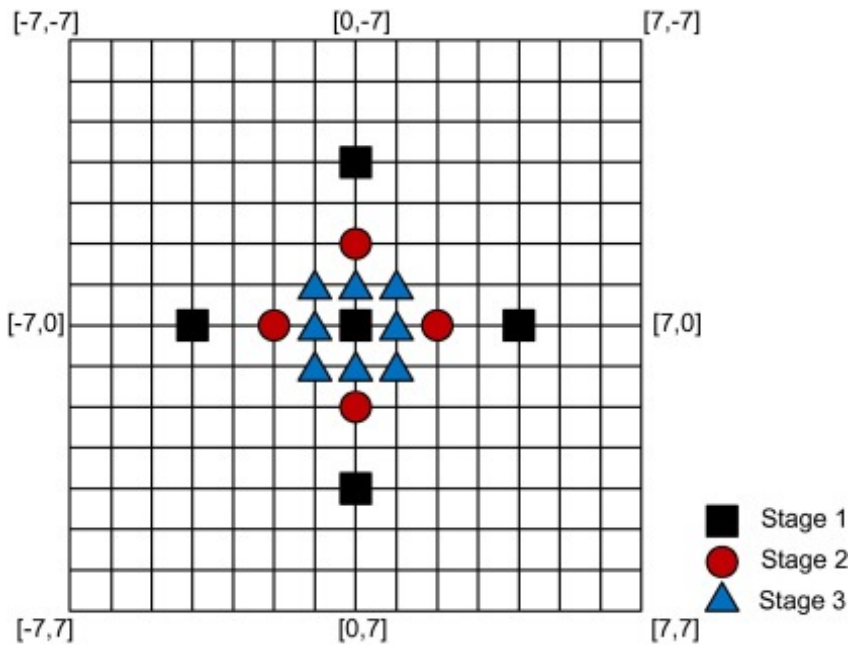
8.4.4。二维对数搜索（TDL）

1981年，??那教和??那教引入了TDL搜索，[12]作为减少被评估的运动矢量候选数量的方法。此方法中使用的搜索模式如图8.18所示，形状为+有五分，但检查九点正方形图案的最后阶段除外。最大的模式最初位于搜索网格的中心，所有五个点都经过评估，并选择BDM最低的点作为下一个搜索的中心。这个过程重复，直到最小点保持在中心，在这种情况下，搜索模式半径减半。这个过程一直持续到半径等于1，这时调用了九点正方形模式。这个过程在算法8.1中描述。

算法8.1 TDL搜索

1. Set initial search radius $r = 2^{k-1}$ where $k = \lceil \log_2(d_m) \rceil$;
2. Set checking points to: $\Gamma = \{[0, 0], [0, \pm r][\pm r, 0]\}$;
3. Evaluate the BDM at the five candidate locations on the cross and select the grid point with the lowest BDM: $\mathbf{d}' = \arg\left(\min_{(i,j) \in \Gamma} (\text{BDM}(i, j))\right)$;
4. IF the minimum remains at the center of the pattern THEN $r = \frac{r}{2}$;
5. IF $r > 1$ THEN $\Gamma = \mathbf{d}' + \{[0, 0], [0, \pm r][\pm r, 0]\}$, GO TO (3);
6. IF $r = 1$ THEN invoke nine-point square pattern with $\Gamma = \mathbf{d}' + \{[0, 0], [\pm r, \pm r], [0, \pm r][\pm r, 0]\}$, $\mathbf{d} = \arg\left(\min_{(i,j) \in \Gamma} (\text{BDM}(i, j))\right)$, STOP.

[下载：下载全尺寸图像](#)

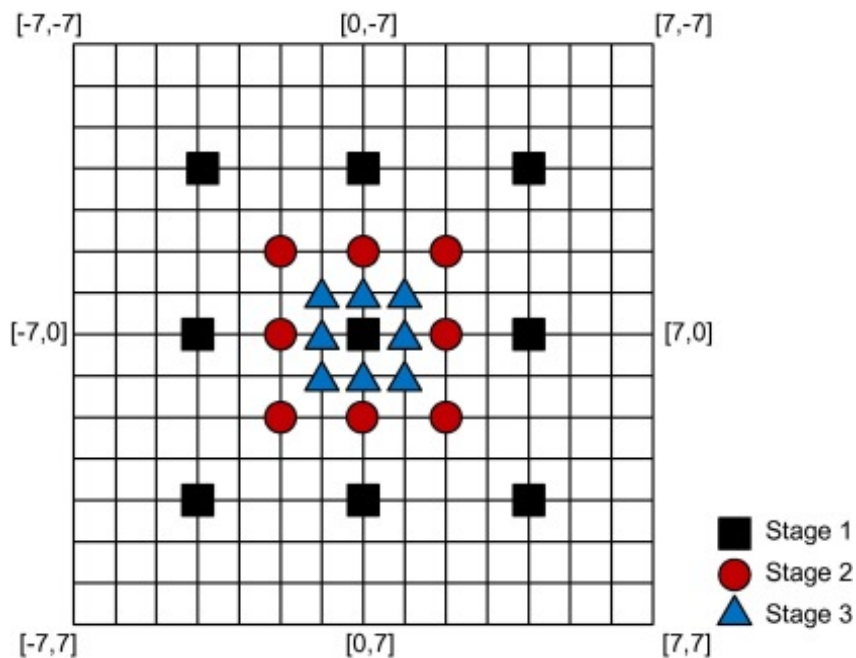


[下载：下载全尺寸图像](#)

图8.18。二维对数搜索模式。

8.4.5。N步搜索 (NSS)

此N-步骤搜索 (NSS) 由Kega等人引入[13]，类似于TDL搜索。如图8.19所示，它使用由九个评估点组成的统一模式，其中每次迭代的搜索半径减少2倍。对于TDL搜索，当搜索半径等于1时，进程终止。这种方法的优点是它需要固定数量的迭代，因此在硬件或软件实现方面提供了规律性。该方法在算法8.2中描述。示例8.4给出了三步搜索 (TSS) 的工作示例。



[下载：下载全尺寸图像](#)

图8.19。N-步骤搜索模式。

NSS算法的复杂性易于计算，是：

$$C_{NSS} = 3 IJF(8k + 1) \quad (8.23)$$

在哪里 k 是算法8.2所示的步骤数。

算法8.2 NSS搜索

1. Set initial search radius $r = 2^{k-1}$ where $k = \lceil \log_2(d_m) \rceil$;
2. Set checking points to: $\Gamma = \{[0, 0], [\pm r, \pm r], [0, \pm r], [\pm r, 0]\}$;
3. Evaluate the BDM at the nine candidate locations and select the grid point with the lowest BDM: $\mathbf{d}' = \arg\left(\min_{(i,j) \in \Gamma} (\text{BDM}(i, j))\right)$;
4. Set $r = \frac{r}{2}$;
5. IF $r < 1$ THEN $\mathbf{d} = \mathbf{d}'$, STOP; ELSE set new search locations: $\Gamma = \mathbf{d}' + \{[0, 0], [\pm r, \pm r], [0, \pm r], [\pm r, 0]\}$; GO TO (3).

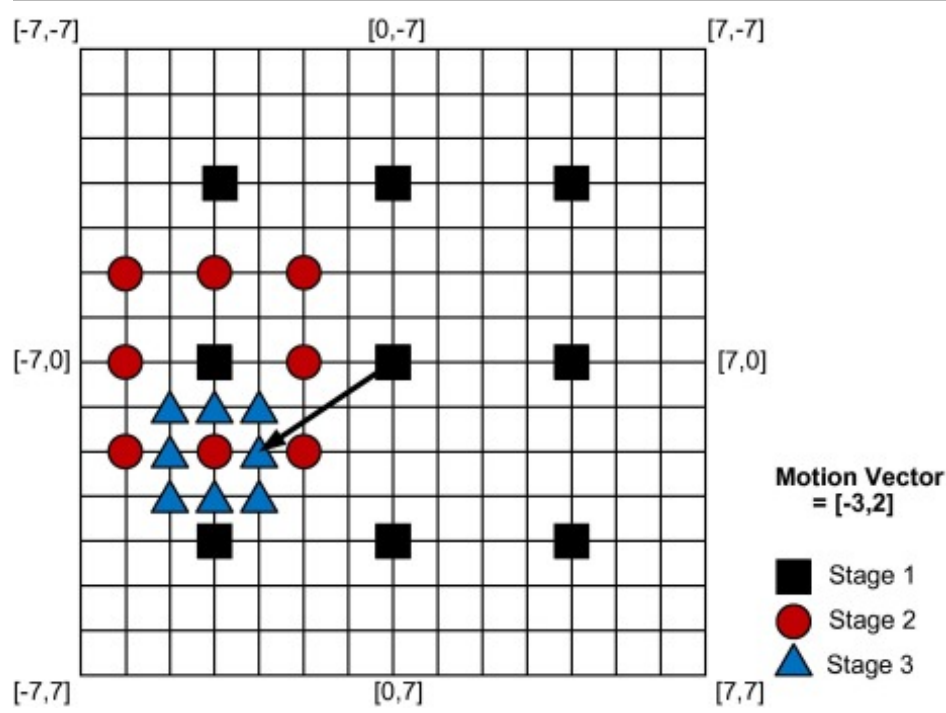
[下载：下载全尺寸图像](#)

示例8.4三步搜索

假设 15×15 搜索网格和具有单个最小值的单调减少误差表面，演示了TSS搜索算法将如何收敛到 $\mathbf{d} = [-3, 2]$ 。

解决方案

NSS算法将始终收敛在 N 步骤，但收敛剖面将取决于误差表面的实际形状。一种可能的解决方案如下图所示。



下载： [下载全尺寸图像](#)

8.4.6。钻石搜索

钻石搜索（DS）由朱和马于1997年首次推出[14]，与TSS和NTSS方法相比，处理速度更快，失真相似。DS采用两种搜索模式——一种大钻石用于一般梯度搜索，另一种小钻石用于最终阶段的精炼。这种方法再次类似于TDL搜索，因为阶段的大型钻石图案 k 以舞台最小BDM为中心 $k-1$ 。当搜索最小值保持在模式的中心时，调用小菱形模式来在终止之前细化运动矢量。这两个钻石图案如图8.20所示，该过程如算法8.3所述。示例8.5给出了钻石搜索过程的示例。应当指出，DS的复杂性取决于最佳运动矢量的方向，内容也取决于内容。

算法8.3 钻石搜索

1. Initialize motion vector estimate: $\mathbf{d}' = [0, 0]$;
2. Set checking points to: $\Gamma = \mathbf{d}' + \{[0, 0], [0, \pm 2], [\pm 2, 0], [\pm 1, \pm 1]\}$;
3. Evaluate the BDM at the nine candidate locations on the diamond and select the grid point with the lowest BDM: $\mathbf{d}' = \arg\left(\min_{(i,j) \in \Gamma} (\text{BDM}(i, j))\right)$;
4. IF the minimum remains at the center of the pattern THEN $\Gamma = \mathbf{d}' + \{[0, 0], [0, \pm 1], [\pm 1, 0]\}$, $\mathbf{d} = \arg\left(\min_{(i,j) \in \Gamma} (\text{BDM}(i, j))\right)$, STOP;
5. GO TO (2).

下载： [下载全尺寸图像](#)

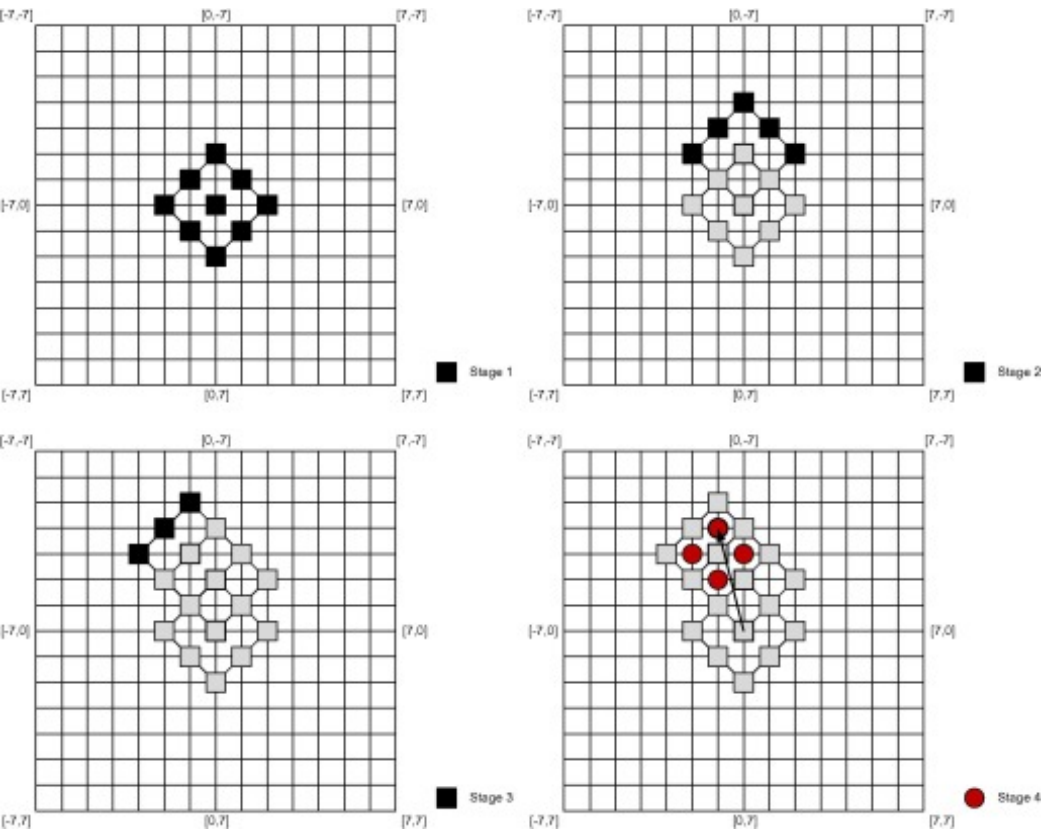
示例8.5钻石搜索

假设 15×15 搜索网格和具有单个最小值的单调递减误差表面，演示了DS搜索算法将如何收敛到

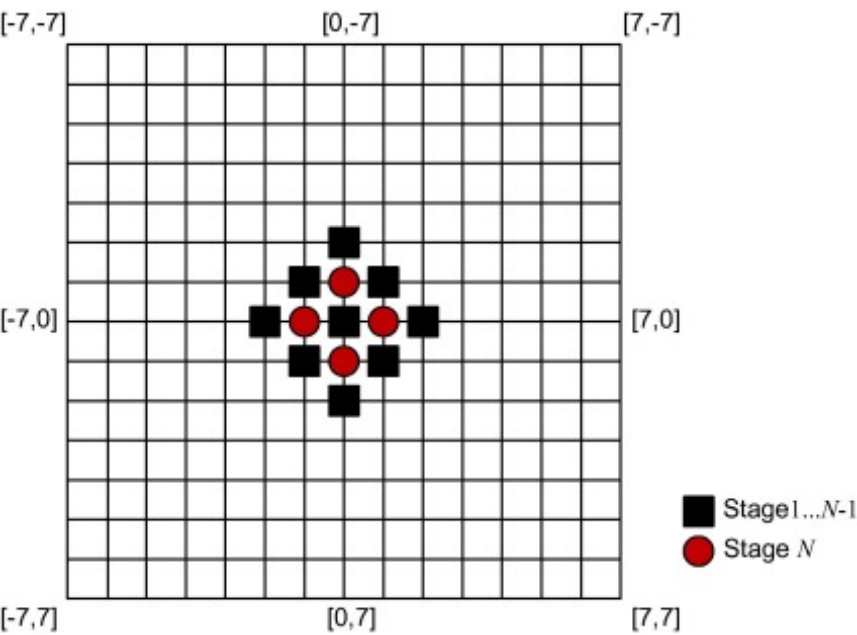
$\mathbf{d} = [-1, -4]$ 。

解决方案

DS算法不会以特定数量的步骤收敛，而是使用大钻石形状跟踪误差表面的梯度，直到最小值停留在钻石中心（本示例中的1-3阶段）。然后在第4阶段引入小钻石形状，以识别最终的运动矢量。收敛剖面当然取决于误差表面的实际形状。一种可能的解决方案如下图所示：



[下载：下载全尺寸图像](#)

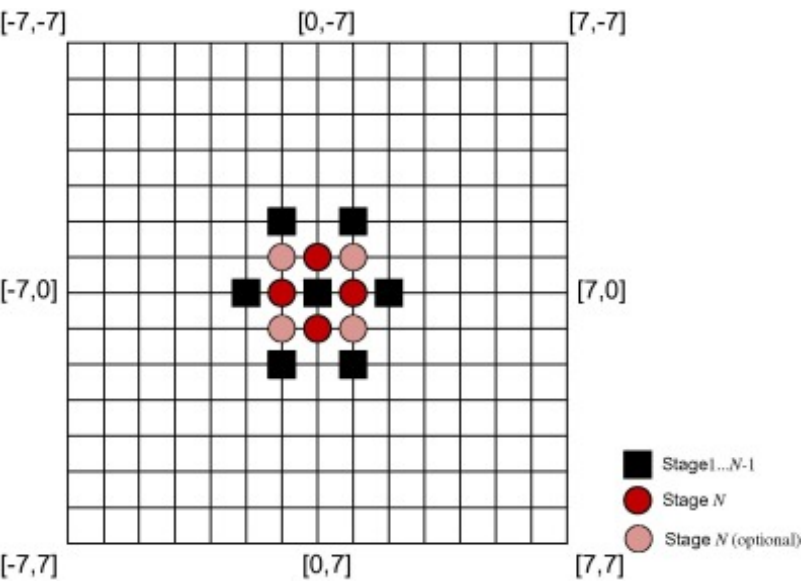


[下载：下载全尺寸图像](#)

图8.20。钻石搜索模式。

8.4.7。六边形搜索

上述方法有很多变体。一个值得注意的是六边形搜索（HEXBS），它再次使用两种模式类型，但这次是六边形的形状（图8.21）。HEXBS由Zhu等人[15]引入，与DS一样，较小的模式在最终迭代中只用于一次细化。HEXBS比钻石搜索具有更大的方向灵活性，在梯度下降阶段的每个迭代中引入的新检查点数是恒定的（三个新点）。



[下载：下载全尺寸图像](#)

图8.21。六边形搜索模式。

DS的前进速度是水平和垂直每步2像素，并且 $\sqrt{2}$ 每一步对角线像素，而HEXBS水平和垂直进步为每一步2像素，水平和垂直 $\sqrt{5}$ 离轴搜索点方向的每一步像素。这意味着HEXBS的搜索性能更加一致，在某些运动矢量位置，其性能可以比DS高出80%。

HEXBS算法的复杂性可以很容易地看出：

$$C_{\text{HEXBS}} = 7 + 3k + 4 \tag{8.24}$$

在哪里 k 是达到误差表面最小值所需的迭代次数。

8.4.8。初始化和终止标准

初始化

在许多情况下，使用智能初始化和终止标准可以提高搜索性能，并有助于避免搜索陷入本地最小

值的问题。参考图8.22，它显示了当前块*P*及其附近的因果邻居*A*“–”*D*，可以根据块运动场的局部平滑度性质，制定一系列候选初始化向量。这些包括：

$$\hat{\mathbf{d}}_P$$
$$= \{[0,0], \mathbf{d}_A, \mathbf{d}_B, \mathbf{d}_C, \mathbf{d}_D, \text{med}(\mathbf{d}_A, \mathbf{d}_C, \mathbf{d}_D), \text{med}(\mathbf{d}_A, \mathbf{d}_B, \mathbf{d}_C)\}$$

(8.25)

<i>B</i>	<i>C</i>	<i>D</i>	
<i>A</i>	<i>P</i>		

下载：下载全尺寸图像

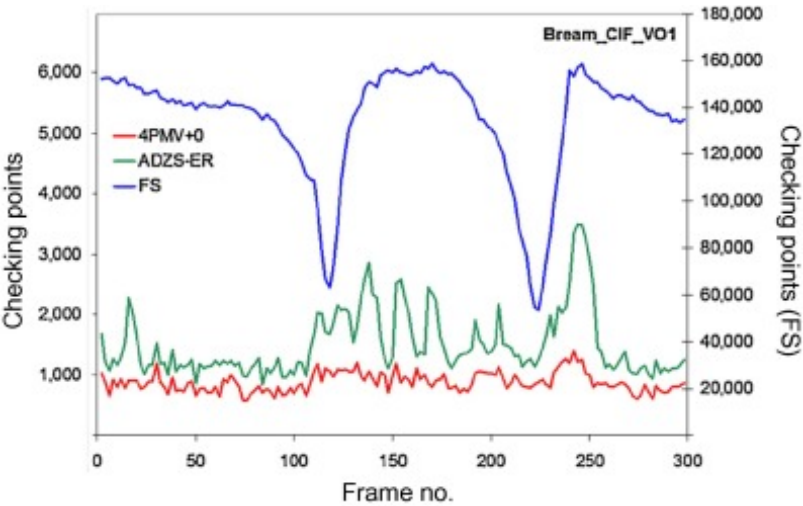
图8.22。用于初始化的预测运动矢量。

初始化算法通常会从方程 (8.25) 中评估部分或全部候选向量，选择BDM最低的向量，然后在 $\hat{\mathbf{d}}_P$ 而不是总是在 $[0,0]$ 。

终止

终止标准的使用在很大程度上取决于所使用的实现方法。例如，在软件实现的情况下，可以通过设置可接受的块失真阈值来终止搜索 BDM_{Th} 并在何时退出搜索 $\text{BDM} \leq \text{BDM}_{\text{Th}}$ 。此外，在每个 SAD评估循环中，当当前偏移量的失真度时，可以终止执行 $[i,j]$ 超过当前最小值。即当 $\text{BDM}_{i,j} \geq \text{BDM}_{\text{opt}}$ ，或者当速率失真成本足够小时。

据报道，几种方法使用智能初始化和终止提供了显著的提速。这些包括图拉皮斯等人[16]、[17]关于PVMFast和EPZS的工作，以及索姆在作者实验室的工作[18]。例如，图8.23显示了复杂性结果，包括4PMV+0算法、钻石搜索和全面搜索。



下载：下载全尺寸图像

图8.23。25 fps的CIF Bream序列的快速运动估计性能比较：4PMV+0 vs ADZS-ER vs FS。

8.4.9。降低复杂度 块匹配算法 性能对比

一些作者发布了表格，从复杂性和预测准确性方面比较了运动估计方法（例如，请参阅参考文献）。[16]）。表8.2比较了本章中介绍的一些算法。可以看出，Simplex Search[19]等方法提供了最准确的结果（最接近FS），具有竞争性复杂性。另一方面，采用智能初始化和终止标准（如4PMV+0）的技术要求极低的检查点数量，但预测准确度略有下降。

表8.2。以25帧/秒的速度降低Bream CIF序列的复杂度BMME比较。顶部：检查点永久阻塞。底部：相对于全搜索的SAD错误。

算法	4PMV + 0	ADZS	DS	单纯形	NSS	FS
最低	1	1	13	9	32.9	1002.4
平均	4.1	5.9	16.1	14.2	33.0	1018.3
最大	9.6	15.3	22.2	21.8	33.1	1024.0
平均	4.248	4.492	4.311	4.078	4.494	3.975
% vs FS	106.9	113.0	108.5	102.6	113.1	100.0

8.5.运动矢量编码

8.5.1。运动矢量预测

H.264/AVC和HEVC等标准在提高速率失真性能方面取得了实质性进展。然而，因此，运动编码开销按比例增加，超过50%的比特用于一些高QP值序列的运动。

通常对运动矢量进行预测编码，以利用块运动场的平滑性。考虑对当前块进行编码P图8.22。可以使用方程 (8.26) 生成预测的运动矢量。如果阻止D没有可用，然后阻止B可以代替。

$$\hat{\mathbf{d}}_P = \{\text{med}(\mathbf{d}_A, \mathbf{d}_C, \mathbf{d}_D)\}$$

(8.26)

然后计算运动矢量差或残差，并编码熵以传输：

$$\mathbf{e}_P = \mathbf{d} - \hat{\mathbf{d}}_P$$

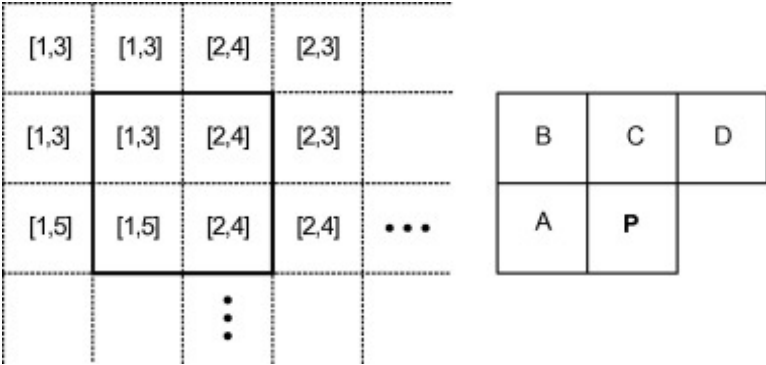
(8.27)

8.5.2。运动矢量的熵编码

运动矢量预测过程将产生一个主要包含小值的装饰相关**残余信号**。就像我们在**第七章**中使用基于霍夫曼的熵编码或算术方法来编码量化变换系数一样，因此我们也可以将其应用于运动向量残差。显然，霍夫曼编码需要不同的VLC表，**算术编码**需要不同的概率分布。

示例8.6运动矢量编码

框架两行中四个相邻块的运动矢量如下所示，粗体概述。使用预测方案： $\hat{\mathbf{d}}_P = \text{med}(\mathbf{d}_A, \mathbf{d}_C, \mathbf{d}_D)$ 计算每个块的预测运动矢量及其残差。



[下载：下载全尺寸图像](#)

解决方案

下表给出了概述的四个块（从左上到右下）的每个块的预测运动矢量和残差：

块, P	1	2	3	4
$\hat{\mathbf{d}}_P$	[1,3]	[2,3]	[1,4]	[2,4]
\mathbf{e}_P	[0,0]	[0,1]	[0,1]	[0,0]

请注意，此处描述的预测过程必须针对边界块进行修改。**第12章**对此进行了进一步描述。

8.6。摘要

本章讨论了运动估计这一重要主题，这是降低图像序列中时间相关性的主要技术。我们看到了如何利用时间相关性来提供额外的编码增益。本章的大部分内容都集中在使用平移运动模型进行块匹配和基于块的运动估计。这种方法被广泛使用，并已在所有标准化视频编解码器中采用。本章还描述了一些降低运动估计复杂性的方法，比较了一系列快速搜索方法以及智能**初始化**和终止技术的描述。

在下一章中，我们将展示如何将**运动估计算法**集成到编码器的DPCM预测循环中。我们还研究了一系列进一步改进这种基于块的混合编解码器编码增益的方法。

参考文献

- [1] T. 易卜拉欣, M. 昆特
多媒体应用可视化数据压缩
IEEE会议记录, 86 (6) (1998年), p.1109-1125
[CrossRef](#) [谷歌学术](#)
- [2] C. 格拉斯贝, K. 马尔迪亚
图像翘曲方法综述
应用统计学杂志, 25 (1998年), p.155-171
[在Scopus中查看记录](#) [谷歌学术](#)
- [3] F. 张, D. 公牛
使用基于区域的纹理模型进行视频压缩的参数框架
IEEE信号处理选题期刊, 6 (7) (2011), p.1378-1392
[CrossRef](#) [在Scopus中查看记录](#) [谷歌学术](#)
- [4] R. 维加斯, A. 卡尔韦, D. 公牛
基于上下文的视频编码
IEEE国际图像处理会议记录 (2013年), p.1953-1957
[CrossRef](#) [在Scopus中查看记录](#) [谷歌学术](#)
- [5] M. Al-Mualla, C. 卡纳加拉贾, D. 公牛
移动通信视频编码
学术出版社 (2002)
[谷歌学术](#)
- [6] A. 内拉瓦利, J. 罗宾斯
运动补偿电视编码: 第1部分
Bell Systems Technical Journal, 58 (1979), p.631-670
[CrossRef](#) [在Scopus中查看记录](#) [谷歌学术](#)
- [7] D. 库格林, D. 海因斯
相位相关图像对齐方法
IEEE控制论与社会国际会议记录 (1975年), p.163-165
[在Scopus中查看记录](#) [谷歌学术](#)
- [8] G. 托马斯, DATV和其他应用的电视运动测量, BBC研究技术报告, 1987/11, 1987年。
[谷歌学术](#)
- [9] Y. 陈, W. 小
用于块矢量运动估计的新型自适应像素抽取
IEEE 视频技术电路和系统交易, 6(1) (1996年), p.113-118
[在Scopus中查看记录](#) [谷歌学术](#)
- [10] B. 刘, A. 扎卡林

运动矢量估计的新快速算法

IEEE 视频技术电路和系统交易, 3 (2) (1993年), p.148-157

[在Scopus中查看记录](#) [谷歌学术](#)

- [11] K. 南, J. 金, R. 帕克, Y. 希姆
一种利用均线金字塔的快速分层运动矢量估计算法
IEEE 视频技术电路和系统交易, 5 (4) (1995), p.344-351
[谷歌学术](#)
- [12] J. 那教, A. 那教
位移测量及其在帧间图像编码中的应用
IEEE通信交易, 29 (12) (1981), 第1799-1808页
[在Scopus中查看记录](#) [谷歌学术](#)
- [13] T. 科加, K. 利努马, A. 平野, T. 石黑
用于视频会议的运动补偿帧间编码
全国电信会议记录 (1981年), p.G5.3.1-G5.3.5
[谷歌学术](#)
- [14] 美国。朱, K. 马
一种新的快速块匹配运动估计的钻石搜索算法
IEEE图像处理事务, 9 (2) (2000年), 第287-290页
[在Scopus中查看记录](#) [谷歌学术](#)
- [15] C.朱, X. 林, L. 周
基于六边形的快速块运动估计搜索模式
IEEE 视频技术电路和系统交易, 12 (5) (2002), p.255-349
[CrossRef](#) [在Scopus中查看记录](#) [谷歌学术](#)
- [16] A.图拉皮斯, O. Au, M. 刘
预测运动矢量场自适应搜索技术 (PMVFAST) : 增强基于块的运动估计
SPIE视觉通信和图像处理记录, 第4310卷 (2001年), 第4310页。883-893
[在Scopus中查看记录](#) [谷歌学术](#)
- [17] A.图拉皮斯
单帧和多帧运动估计的增强预测分区搜索
SPIE视觉通信和图像处理会议记录, 第4671卷 (2002年), 第4671页。1069-1078
[CrossRef](#) [谷歌学术](#)
- [18] O. Sohm, 运动矢量估计方法, 美国专利7,260,148, 2007年。
[谷歌学术](#)
- [19] M.Al-Mualla, C.卡纳加拉贾, D. 公牛
用于单引用和多引用运动估计的单纯形最小化
IEEE 视频技术电路和系统交易, 11 (12) (2001), pp.1209-1220
[在Scopus中查看记录](#) [谷歌学术](#)

- 1 形式上，对于表观运动，位移场被称为对应场。
- 2 在不失去通用性的情况下，我们在这里使用相邻的框架。
- 3 注：SAD通常优先用于平均绝对差（MAD）。

版权所有?2014爱思唯尔有限公司。保留一切权利。



[关于ScienceDirect](#)

[远程访问](#)

[购物车](#)

[广告](#)

[联系和支持](#)

[条款和条件](#)

[隐私政策](#)



我们使用cookie来帮助提供和增强我们的服务，并定制内容和广告。继续即表示您同意**使用cookie**。

版权所有?2021爱思唯尔B.V.或其许可方或贡献者。ScienceDirect ®是爱思唯尔B.V.的注册商标。

ScienceDirect ®是爱思唯尔B.V.的注册商标。