

一种基于 H. 264 的有效运动估计算法

段哲民¹, 周 巍¹, 李 辉¹, 周 欣²

(1. 西北工业大学电子信息学院, 陕西西安 710072; 2. 西北工业大学计算机学院, 陕西西安 710072)

摘 要: 运动估计算法是实时视频编解码技术的研究重点, 高精度的匹配和补偿可以减少预测误差, 提高视频图像的压缩效果. 为降低在视频编码标准 H. 264 中运动估计的高计算复杂度问题, 提出了采用一种基于节点模型的可变形块匹配运动估计算法来搜索最佳运动矢量. 该算法充分利用了 H. 264 运动矢量的统计特性和相关性, 并采用基于像素差值分类的运动估计匹配准则. 实验表明, 在编码性能损失很小的条件下, 该算法有效降低了视频压缩编码中运动估计的运算复杂度.

关键词: 视频编码; 运动估计; 块匹配; 匹配准则

中图分类号: TN919.81 **文献标识码:** A **文章编号:** 0372-2112 (2006) 12A-2406-04

An Efficient Motion Estimation Algorithm for H. 264 Video Coding Standard

DUAN Zhe-min¹, ZHOU Wei¹, LI Hui¹, ZHOU Xin²

(1. College of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China;

2. Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China)

Abstract: The efficient motion estimation algorithms are the important research hotspot in the real-time video coding-decoding technology. The prediction error can be decreased by incorporating high accuracy estimation and compensation, and the performance of compressed video can be improved. The high computational demands of the H. 264 video encoder are reduced by an deformable block matching algorithm using the node-based model. This algorithm well exploits the motion vector statistic characteristic and correlation and uses motion estimation criterion based on the pixel difference classify. Experiments show that the efficient motion estimation algorithm proposed can further reduce block matching distortion computations relative to conventional fast motion estimation algorithms with neglectable coding efficiency loss.

Key words: video coding; motion estimation; block-matching; matching criterion

1 引言

基于块的运动估计和补偿是视频编码中最通用的算法, 它把图像域分割成互不重叠的称为块的小区域, 并且假定每一个块内的运动都可以用一个简单的参数模型特征化. 如果块足够小, 那么这种模型是相当合理的, 而且这种方法被广泛用于视频标准变换运动补偿滤波和采用基于块的运动补偿进行的数字视频压缩. 块匹配算法由于具有较少的硬件复杂度, 容易在超大规模集成电路中实现, 因此是最通用的运动估计算法.

本文在深入研究 H. 264 视频编码标准及其运动矢量统计特征的基础上, 提出了采用一种基于节点模型的可变形块匹配运动估计算法. 实验结果表明, 本文的算法不仅保证了编码器原有的率失真特性, 而且大大降低了运算复杂度与实现上的难度.

2 基于 H. 264 的块匹配运动估计算法分析

将图像序列的每一帧分成多个大小为 $M \times N$ 的宏块(一般情况下 $M=N$), 然后对于当前帧中的每一块根据一定的匹配准则在参考帧某一给定搜索范围内找出与当前块最相似的块, 即匹配块, 由匹配块与当前块的相对位置计算出运动位移, 所得运动位移即为当前块的运动矢量. 运动估计越准确, 补偿的残差就越小, 编码的效率也就越高, 解码出的图像质量越好.

最新视频压缩标准 H. 264 是由 ISO/IEC 的运动图像专家组(MPEG)和 ITU-T 的视频编码专家组(VCEG)共同制定, 其主要特点是具有更高的编码效率和更好的网络适应性. 在相同重构图像质量下, 由于 H. 264 预测精度达到了 $1/4$ 像素, 与 H. 263 和 MPEG-4 标准相比能节约 50% 的码流, 但是所带来的结果是, 运算的复杂度是 H. 263 编码复杂度的 4 倍. 运动估计

模块是整个编码模块中运算量最大的部分,在JVT的文档中表明运动估计在使用一个参考帧时占整个运算量的70%,在使用5个参考帧时占整个运算量的90%,因此需要提高运动估计的效率,使运动估计算法的搜索过程更健壮、更迅速、更高效。全搜索算法以其极高的精度受到广泛的关注,但由于其要求对搜索窗内所有点进行匹配,所需运算量十分巨大,很难适应实际应用,特别是实时应用的要求。为了追求较高运动估计精度下的较低运算量,人们不断地提出各种快速算法。概览这些快速算法,大致可分为以下几类:(1)通过在搜索窗内按一定规则,仅搜索较少的几个点来降低运算复杂度,经典算法有3步法(3SS)、新3步法(N3SS)^[1]、4步法(4SS)^[2]等;(2)通过使用相对较为简单的匹配准则来降低运算复杂度,如最小最大误差等;(3)通过利用代数不等式排除不可能成为最佳匹配的待搜索点,从而减少匹配点数,降低运算复杂度,如近两年提出的一些快速全搜索算法等。本文在深入研究H. 264视频编码标准及其运动矢量统计特征的基础上,提出了一种基于节点模型的可变形块匹配运动估计算法。

块匹配的运动估计算法的基本思想是依据一定的匹配准则,通过在两帧之间的像素域利用搜索程序找到最佳的运动矢量估计。如图1所示,其中帧 k (当前帧)中 $M \times N$ 像素块(中心位置在 (x_0, y_0))的位移就是通过搜索帧 $k-1$ 中同样大小的最佳匹配块来确定。从计算因素考虑,搜索通常限制在 $(M+2M_1, N+2N_1)$ 范围内。 M_1, N_1 的值可以根据具体的估计要求来确定。

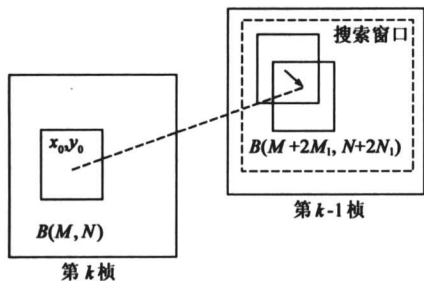


图1 块匹配法原理

在块匹配算法中采用不同的匹配准则则会获得不尽相同的运动估计结果。采用均方差(MSD)准则的匹配效果优于绝对帧差准则(MAD),但是在H. 264视频编解码系统中,帧间编码是对运动补偿差信号进行编码,所以MAD的计算次数和复杂度将少于MSD,但会在一定程度上牺牲运动估计的性能。基于H. 264视频编解码的特点,本文采用像素差值分类的运动估计匹配准则。

3 H. 264/AVC运动矢量统计特征分析

在视频序列中,视频对象的运动具有连续的运动特征,因此描述视频序列运动特征的宏块的运动矢量间也存在时间和空间上的相关性。而且由于视频序列运动程度的不同,相邻宏块间运动相关性存在非常明显的变化。这一变化对H. 264/AVC编码结果的影响可以通过对不同视频序列的宏块编码模式进行的统计来看^[3],如表1所示。

可以发现对于运动平滑的视频序列(Foreman序列)或视

频图像中的平坦区域,宏块大都采用了大编码模式,即 16×16 、 16×8 和 8×16 模式;而对于运动剧烈的视频序列Stefan,编码时更多的宏块则采用了小编码模式,即 8×8 、 8×4 、 4×8 和 4×4 模式。如图2所示。

表1 不同视频序列宏块编码模式的比较

视频序列	大模式宏块的个数	大模式宏块的比例
Stefan(300帧)	53906	45.38%
Foreman(150帧)	44651	75.17%

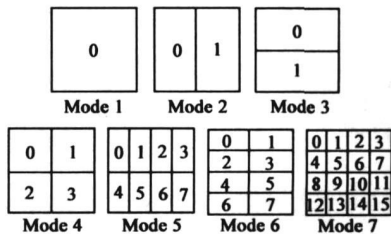


图2 7种预测块模式

H. 264/AVC采用的7种分块模式可以在视频序列不同的运动程度情况下选择使用,而且可以随着运动程度的变化,模块的选择也作相应的调整。基于上述分析,本文提出采用一种基于节点模型的可变形块匹配运动估计算法。

4 基于节点模型的运动估计算法

4.1 基于节点的运动表示

在基于节点的块运动模型中,假设在块中所选择的某个数目的控制节点能够自由运动,而且任何中间点的位移可以通过节点的位移内插。令 K 表示控制节点数目, B_m 代表第 m 个图像块, $d_{m,k}$ 表示 B_m 中控制节点的运动矢量(MV),则块的运动函数可以描述为:

$$d_m(x) = \sum_{k=1}^K \phi_{m,k}(x) d_{m,k}, \quad x \in B_m \quad (1)$$

式(1)表示作为节点位移的内插块中任意点的位移^[4]。内插核 $\phi_{m,k}(x)$ 取决于 B_m 中第 k 个控制节点所期望的对 x 的贡献。一般使用与相应的节点结构有关的行状函数来设计这种内核。如果采用多节点的模型则可以表征在运动过程中比较复杂的变形。在单节点情况下(在块中心或一个选择的角)的内插核是一个脉冲函数,相当于最临近内插。在三节点(块的任意三个角)和四节点(四个角)情况下的内插函数分别是仿射函数和双线性函数。要对矩形块应用仿射模型,首先要将矩形分成两个三角形,然后每个三角形模型可以化为三节点模型。

当块的四个角点的位置给定后就可以估计出节点的MV,根据关于运动的动态范围和期望估计精度的先验知识可以确定出节点MV合适的搜索范围和搜索步长。而且由于基于节点模型表示中所有的运动参数同等重要,采用整数或半像素精度就可以规定节点的MV。

4.2 采用基于节点模型的运动估计算法^[4]

对于任意图像块 B_m ,在基于节点的运动模型中,任何块的运动参数是节点的MV,即 $a = [d_k, k \in K]$,其中 $K = \{1, 2, \dots, K\}$ 。节点的MV可以通过使这个块上的预测误差最小化来进行估计,即:

$$E(\mathbf{a}) = \sum_{\mathbf{x} \in B} |\Psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a})) - \Psi_1(\mathbf{x})|^P \quad (2)$$

$$\text{其中 } \mathbf{w}(\mathbf{x}; \mathbf{a}) = \mathbf{x} + \sum_{k \in K} \phi_k(\mathbf{x}) \mathbf{d}_k \quad (3)$$

由于采用穷尽搜索算法需要花费较大的计算量, 因此采用基于梯度的搜索算法来最小化式(2)中的误差^[5].

定义 $\mathbf{a} = [\mathbf{a}_x^T, \mathbf{a}_y^T]^T$, 其中 $\mathbf{a}_x = [d_{x,1}, d_{x,2}, \dots, d_{x,k}]^T$, $\mathbf{a}_y = [d_{y,1}, d_{y,2}, \dots, d_{y,k}]^T$, 可得:

$$\frac{\partial E}{\partial \mathbf{a}}(\mathbf{a}) = \begin{bmatrix} \frac{\partial E}{\partial \mathbf{a}_x}(\mathbf{a}) \\ \frac{\partial E}{\partial \mathbf{a}_y}(\mathbf{a}) \end{bmatrix} \quad (4)$$

$$\text{其中 } \frac{\partial E}{\partial \mathbf{a}_x}(\mathbf{a}) = 2 \sum_{\mathbf{x} \in B} e(\mathbf{x}; \mathbf{a}) \frac{\partial \Psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a}))}{\partial \mathbf{x}} \phi(\mathbf{x}) \quad (5)$$

$$\frac{\partial E}{\partial \mathbf{a}_y}(\mathbf{a}) = 2 \sum_{\mathbf{x} \in B} e(\mathbf{x}; \mathbf{a}) \frac{\partial \Psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a}))}{\partial \mathbf{y}} \phi(\mathbf{x}) \quad (6)$$

在式(5)、(6)中, $e(\mathbf{x}; \mathbf{a}) = \Psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a})) - \Psi_1(\mathbf{x})$, $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_k(\mathbf{x})]^T$.

舍去涉及二阶梯度的项, 黑塞矩阵可以近似为:

$$[\mathbf{H}(\mathbf{a})] = \begin{bmatrix} \mathbf{H}_{xx}(\mathbf{a}) & \mathbf{H}_{xy}(\mathbf{a}) \\ \mathbf{H}_{yx}(\mathbf{a}) & \mathbf{H}_{yy}(\mathbf{a}) \end{bmatrix} \quad (7)$$

$$\text{其中 } [\mathbf{H}_{xx}(\mathbf{a})] = 2 \sum_{\mathbf{x} \in B} \left[\frac{\partial \Psi_2}{\partial \mathbf{x}} \right]_{\mathbf{w}(\mathbf{x}; \mathbf{a})}^2 \phi(\mathbf{x}) \phi(\mathbf{x})^T \quad (8)$$

$$[\mathbf{H}_{yy}(\mathbf{a})] = 2 \sum_{\mathbf{x} \in B} \left[\frac{\partial \Psi_2}{\partial \mathbf{y}} \right]_{\mathbf{w}(\mathbf{x}; \mathbf{a})}^2 \phi(\mathbf{x}) \phi(\mathbf{x})^T \quad (9)$$

$$[\mathbf{H}_{xy}(\mathbf{a})] = 2 \sum_{\mathbf{x} \in B} \frac{\partial \Psi_2}{\partial \mathbf{x}} \frac{\partial \Psi_2}{\partial \mathbf{y}} \Big|_{\mathbf{w}(\mathbf{x}; \mathbf{a})} \phi(\mathbf{x}) \phi(\mathbf{x})^T \quad (10)$$

因此得到梯度的更新算法为:

$$\mathbf{a}^{(l+1)} = \mathbf{a}^{(l)} - \alpha [\mathbf{H}(\mathbf{a}^{(l)})]^{-1} \frac{\partial E}{\partial \mathbf{a}}(\mathbf{a}^{(l)}) \quad (11)$$

在每次迭代的更新中都需要 $2K \times 2K$ 对称矩阵 $[\mathbf{H}]$ 的求逆. 为了减少运算复杂度, 可以分别更新 x 和 y 方向上的位移. 通过类似的推导可以得到

$$\mathbf{a}_x^{(l+1)} = \mathbf{a}_x^{(l)} - \alpha [\mathbf{H}_{xx}(\mathbf{a}^{(l)})]^{-1} \frac{\partial E}{\partial \mathbf{a}_x}(\mathbf{a}^{(l)}) \quad (12)$$

$$\mathbf{a}_y^{(l+1)} = \mathbf{a}_y^{(l)} - \alpha [\mathbf{H}_{yy}(\mathbf{a}^{(l)})]^{-1} \frac{\partial E}{\partial \mathbf{a}_y}(\mathbf{a}^{(l)}) \quad (13)$$

在这种情况下, 每次更新只需要两个 $K \times K$ 矩阵的求逆就可以. 在四节点情况下, $[\mathbf{H}]$ 是一个 8×8 的矩阵, 而 $[\mathbf{H}_{xx}]$ 和 $[\mathbf{H}_{yy}]$ 是 4×4 的矩阵, 从而减少了运算复杂度. 本算法的初始值由穷尽搜索块匹配算法提供, 考虑到四节点模型, 每一个块的每一个角都是一个节点, 可以用与每个节点相连的四个块的运动矢量的平均作为这个节点 MV 的初始估计, 然后这个初始估计可以由式(11)进行迭代更新.

4.3 搜索中匹配准则的选择

在块匹配算法中采用不同的匹配准则会获得不尽相同的运动估计结果. 运动估计算法中常用的匹配准则有最小绝对差值(MAD)、最小均方误差(MSE)和归一化互相关函数(NCCF). 分别的定义为:

(1) 最小绝对值误差

$$MAD(d_x, d_y) = \frac{1}{MN} \sum_{(x_1, y_1) \in B} |f_k(x_1, y_1) - f_{k-1}(x_1 + d_x, y_1 + d_y)| \quad (14)$$

(2) 最小均方误差

$$MSE(d_x, d_y) = \frac{1}{MN} \sum_{(x_1, y_1) \in B} [f_k(x_1, y_1) - f_{k-1}(x_1 + d_x, y_1 + d_y)]^2 \quad (15)$$

(3) 归一化互相关函数

$$NCCF(d_x, d_y) = \frac{\sum_{(x_1, y_1) \in B} f_k(x_1, y_1) f_{k-1}(x_1 + d_x, y_1 + d_y)}{\left[\sum_{(x_1, y_1) \in B} f_k^2(x_1, y_1) \right]^{1/2} \left[\sum_{(x_1, y_1) \in B} f_{k-1}^2(x_1 + d_x, y_1 + d_y) \right]^{1/2}} \quad (16)$$

若在某一点 (x, y) 处的 MAD 、 MSE 值最小或 $NCCF$ 值最大, 则该点为最优匹配点; 在 H. 264 视频编解码系统中, 帧间编码是对运动补偿差信号进行编码, 而 MAD 准则不需要作乘法运算, 所以 MAD 的计算次数和复杂度将少于 MSE 和 $NCCF$, 所以使用最多. 通常用 SAD 代替 MAD , SAD 即求绝对误差和, 其定义如下:

$$SAD(d_x, d_y) = \sum_{(x_1, y_1) \in B} |f_k(x_1, y_1) - f_{k-1}(x_1 + d_x, y_1 + d_y)| \quad (17)$$

分别求出当前块和其相邻块间的 SAD 值, 然后选取 SAD 值最小的块的运动矢量作为预测值. 利用 SAD 准则预测精度高, 但计算 SAD 值的时间开销大, 因此利用运动矢量的相关性以及 H. 264 视频编解码的特点, 本文采用像素差值分类的运动估计匹配准则.

该准则将参考像块中的每一个像素分类为匹配像素和失配像素进行统计, 在运动搜索中选取参考块中失配像素最少的块作为匹配块, 可以对保持当前块与匹配块形状的一致性有较强的约束^[6]. 具体运算过程如下:

设 $P_f(u, v)$ 为当前帧中的像素, $P_{f-1}(u + i, v + j)$ 为参考帧 $f-1$ 中相对于 (u, v) 点水平、垂直分别移动了 i, j 的像素, 对于选定的门限 t , 进行如下分类:

$$T(u, v, i, j) = \begin{cases} 1, & \text{若 } |P_f(u, v) - P_{f-1}(u + i, v + j)| \leq t \\ 0, & \text{其他} \end{cases} \quad (18)$$

其中“0”表示该像素对位移为 i, j 失配, “1”表示 $P_{f-1}(u + i, v + j)$ 是 $P_f(u, v)$ 的匹配像素. 在整个块内对 T 值进行求和运算:

$$S(i, j) = \sum T(u, v, i, j) \quad (19)$$

$S(i, j)$ 表示当前块与位移为 i, j 的匹配块中的匹配像素数. 在规定的搜索范围内, 选取 $S(i, j)$ 最大的块为最佳匹配块, 即:

$$S_m(dh, dv) = \max(S(i, j)) \quad (20)$$

其中 dh, dv 为该块的水平和垂直位移矢量. 因此可以看到采用这种匹配准则的运动估计体现了最小风险准则, 在运动搜索中选取参考块中失配像素最少的块作为匹配块, 对保持当前块与匹配块形状的一致性有较强的约束. 但在运动搜索范围内, 有些位置在本匹配准则下会具有相同的匹配像素数, 即

有相同的风险. 对于此种情况再采用基于最小相对位移约束的判决方法进行二次匹配.

设当前块为 $B_r(k, l)$, 包含以 (k, l) 为起点的 $m \times n$ 个像素, 相邻的前一块为 $B_{r-1}(k_1, l_1)$, 同样包含以 (k_1, l_1) 为起点的 $m \times n$ 个像素, B_{r-1} 块的运动矢量为 $dh(r-1), dv(r-1)$. 块匹配运动估计的搜索范围为水平 $i \in [-p, p]$, 垂直 $j \in [-q, q]$. 根据上述匹配准则的定义求出 $B_r(k, l)$ 在参考帧上对 i, j 位移块的匹配像素数 $S(i, j)$. 若有位移值 \ddot{u}, \ddot{j} 则使:

$$S(\ddot{u}, \ddot{j}) = \max(S(i, j)) \tag{21}$$

当且仅当有唯一的 \ddot{u}, \ddot{j} 使上式成立时, 位移矢量:

$$dh(r) = \ddot{u}, dv(r) = \ddot{j} \tag{22}$$

若有 $\{i_x, j_x\}, x = 1, 2, \dots$ 同时满足时, 则计算:

$$\Delta d(i_x, j_x) = |dh(r-1) - i_x| + |dv(r-1) - j_x| \tag{23}$$

$$\Delta d(i_opt, j_opt) = \min(\Delta d(i_x, j_x)) \tag{24}$$

选取 $\Delta d(i_opt, j_opt)$ 与对应的位移矢量 (i_opt, j_opt) 为最佳运动矢量. 因此在上述匹配准则中, 若有多个不同位移的参考块有相同最小风险时, 选择与相邻像块相对位移最小的像块为当前块的最佳匹配块. 运用这样的判断准则不但显著的提高了判决的准确性和唯一性, 而且又由于在相同风险下选择最小的相对位移, 使得相邻图像区域的运动矢量场的一致性得到进一步的提高, 减小了搜索像素的复杂度.

5 实验结果

在 JVT 的 H. 264/ AVC 编码参考模型 JM8.0 上实现本文所提算法. 选择具有广泛代表性的 6 个标准测试序列: Bus, Carphone, Claire, Foreman, Flower, Stefan. 其中, Stefan 为大运动序列, Carphone, Foreman, Bus 为中等运动序列, Claire 为小运动或几乎静止序列, Flower 则包含了较多的细节与镜头的平移. 采用的帧率是 30Hz, 搜索范围是 $[-32, 32]$, 5 个参考帧, CABAC 熵编码, 使用全部 7 种块进行匹配. 为便于比较, 还实现了 3 个经典算法: FS (full search algorithm)、N3SS 和 MVFAST (Motion Vector Field Adaptive Search Technique)^[7] 算法.

表 2 平均峰值信噪比 PSNR/ dB

	FS	N3SS	MVFAST	本文算法
Stefan	35.28	35.04	35.17	35.22
Flower	37.33	36.97	37.19	37.28
Foreman	36.95	36.81	36.84	36.87
Bus	39.83	39.69	39.70	39.77
Claire	36.92	36.71	36.74	36.82
Carphone	39.95	39.80	39.81	39.86

表 2 列出了 6 种视频序列的平均 PSNR 值, 可以看出, 对于“Stefan”视频序列, 本文算法的 PSNR 值为 35.22dB, 在相同的比特率下, 与 MVFAST 算法相比, PSNR 值提高了 0.05dB. 对于“Bus”视频序列, 本文算法的 PSNR 值为 39.77dB, 与 MVFAST 算法相比, PSNR 值提高了 0.07dB; 与 N3SS 算法相比, PSNR 值提高了 0.09dB. 而且本文算法与 FS 算法相比, 视频序列的平均 PSNR 值下降也非常小.

表 3 列出了本文算法与 FS 算法、N3SS 算法和 MVFAST 算法的运动估计搜索时间的对比. 可以看出, 对“Flower”视频序

列, 本文算法与 N3SS 算法相比平均运动估计速率提高了 59.23%, 与 MVFAST 算法相比平均运动估计速率提高了 19.14%; 对“Carphone”视频序列, 与 N3SS 算法相比平均运动估计速率提高了 62.45%, 与 MVFAST 算法相比平均运动估计速率提高了 20.57%, 可见本文采用的算法在加速运动估计速度的同时, 提高了编码的实时性.

表 3 运动估计搜索时间的对比/s

	FS	N3SS	MVFAST	本文算法
Stefan	170.4	0.608	0.473	0.453
Flower	171.1	0.500	0.455	0.330
Foreman	165.7	0.600	0.585	0.443
Bus	162.4	0.612	0.540	0.360
Carphone	188.3	0.549	0.473	0.405
Claire	187.7	0.318	0.250	0.245

6 结论

本文提出了一种适用于最新视频编码标准 H. 264/ AVC 的块匹配运动估计算法. 该算法针对 H. 264 的编码特点, 利用视频图像序列的运动矢量统计特性以及时域、空域的相关性, 采用基于节点模型的可变形块匹配运动估计算法以及基于像素差值分类的运动估计匹配准则. 实验结果表明, 在保持 PSNR 方面性能不受损失的同时, 大大降低了运算复杂度.

参考文献:

[1] Joint Video Specification (ITU-T Rec. H. 264/ISO/IEC 14496-10 AVC)-Joint Committee Draft, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG[C]. 7th Meeting: Pattaya, Thailand, 2003. 7- 14.

[2] Li Renxiang, Zeng Bing, Liu Ming L. A three step search algorithm for block motion estimation[J]. IEEE Transaction on Circuits System Video Technology, 1994, 4(4): 438- 442.

[3] Xue Jir zhu, Shen Lar sun. An efficient block matching motion estimation algorithm for H. 264/ AVC[J]. Acta Electronic Sinica, 2004, 32(4): 583- 586. (in Chinese)

[4] Yao Wang, Jom Ostemann, Ya Qin Zhang. Video Processing and Communications[M]. Beijing: Publishing House of Electronics Industry, 2003: 154- 169.

[5] Lee O, Y Wang. Motion compensated prediction using nodal based deformable block matching[J]. Journal of Visual Communication and Image Representation. 1995, 6(1): 26- 34.

[6] He Gu ming, Wu Ba yuan et al. The Technology of Video Coding and Communications Control Based on Object[M]. Wuhan: Publishing House of Wuhan University, 2005. 177- 188.

[7] Hosur P I, Ma K K. Motion vector field adaptive fast motion estimation[A]. Proc 2nd Int Conf Information, Communications and Signal Processing (ICICS 99)[C]. Singapore, 1999. 7- 10.

作者简介:

段哲民 男 1953 年 8 月出生于陕西, 现为西北工业大学电子信息学院教授, 研究方向为电路与系统, 雷达信号处理等.
E-mail: zhemin@nwpu.edu.cn