一种低复杂度的基于进化策略的自适应运动估计方法

王 辉 毛志刚
(哈尔滨工业大学微电子中心,哈尔滨 150001)

摘 要 为了进一步提高编码质量并能快速编码,提出了一种新的基于进化策略的自适应运动估计算法。鉴于在进化策略中变异操作与正态分布法则对应,是核心算子,为此将进化策略应用于运动估计,提出了一种新的自适应运动估计算法,并第 1 次将运动方向信息作为变量引入运动估计算法,同时改进了步长自适应控制机制,以便进一步提高算法的收敛速率,同时采用种群规模的自适应控制,降低了算法的复杂度。试验结果表明,该算法的性能与全搜索算法相近,而复杂度略大于三步法。由于其具有低复杂度和进化算法的内在并行性的特点,故该算法适合硬件实现。

关键词 运动估计 相关变异 进化策略 1/5 规则
中图法分类号: TN919.81 **文献标识码**: A **文章编号**: 1006-8961(2005)07-0878-06

A Low Complexity Adaptive Motion Estimation Algorithm Based on Evolution Strategies

WANG Hui, MAO Zhi-gang
(*Harbin Institute of Technology Microelectronics Center, Harbin* 150001)

Abstract Based on evolution strategies(ESs) a novel adaptive motion estimation search algorithm is presented in order to improve the encoder quality. The mutation operator in ESs is used with frequency corresponding to the normal distribution law. This algorithm applies the ES algorithm to block motion estimation, the motion direction factor participates in motion vector computing as a variable for the first time, and affects the whole search process, neither just being an implicit factor nor a predictive measure. The adaptive schemes are advanced in the step length control and population sizing. Experimental results demonstrate that this algorithm has similar performance to that of the full search algorithm, and owing to its inherent parallelism and low complexity it is suitable for VLSI implementation.

Keywords motion estimation, correlated mutation, evolution strategies, 1/5 success rule

1 引 言

在视频编码系统中,如 H. 263, MPEG-4 和 AVC,运动估计是编码算法中的核心模块,由于它计算消耗最大,约占编码器计算量的 60% ~80%,因此在给定的位率下,运动估计的结果对图像编码质量有很大影响。

目前,应用最普遍的运动估计算法是块匹配算法,而在块匹配算法中,精度最高的则是全搜索算法,但全搜索算法的高运算复杂度使它很难被应用在实时性要求高的视频系统中。为降低运算复杂度,很多快速算法被提出来,如三步法、菱形搜索法、预测算法等。这些算法虽大大降低了运算的复杂度,但它们只是对某些特定点进行检测,而且这些快速算法都是基于单调性假设,即"匹配误差与搜索点到最优点的距离成正比",由于这一假设不完全符合实际情况,且通常状况下,匹配误差在搜索区域

收稿日期:2004-05-14;**改回日期**:2004-12-13

第一作者简介:王辉(1975 ~),男。1999 年获哈尔滨工业大学微电子学与固体电子学硕士学位,现为哈尔滨工业大学微电子中心博士研究生。主要从事 VLSI 设计及视频编码算法研究。E-mail:wangh@china.com

内呈多峰值状态,因此这些快速算法易于陷入局部最优。

为避免由于单调性假设而导致的性能损失,一些基于遗传算法的快速搜索算法被提出来[1-3],但遗传算法的特性要求必须有足够的个体数量和循环代数才能得到优化的结果,在已经提出的算法中,种群中个体数至少为 16,这使得基于遗传算法的运动估计算法很难减小计算复杂度。

遗传算法和进化策略(evolution strategies,简称 ESs)都是进化算法的主要分支,它们紧密相关,但又各自独立发展。遗传算法是在染色体层次上进行优化,并注重仿真基因遗传机制;进化策略则是在表现型的基础上考虑进化过程,在进化策略中,一般只采用选择和变异算子,而且其变异算子与正态分布法则对应。对于数值优化问题,进化策略不仅比传统的遗传算法优化速度快,而且更可能找到真正的全局最优解[4]。

在已发表的基于进化算法的运动估计算法中,并未将运动方向信息作为一个变量参与进化操作,或者只是作为一个强制性的加速因子[3],而在本文提出的算法中,运动方向信息则作为一个变量引入,与 x,y 变量同样参与进化操作。

2　进化策略的基本原理

进化策略由 Rechenberg 在 1964 年提出,Schwefel 对其进行了更进一步的发展[4]。这种方法最初被用来解决复杂的离散或连续参数的优化问题。在进化策略中,主要的进化操作是变异,也可以进行重组操作。

基本的进化策略算法是 (μ,λ)-ES,其中 $\lambda \geqslant \mu \geqslant 1$。$(\mu,\lambda)$ 表示在一代中,μ 个父亲产生 λ 个子孙。这里只考虑简化的进化策略算法,即不包含重组操作的进化策略算法。

一个全局优化问题可以被描述为一个二元组 $(\boldsymbol{S},f)$,其中 $\boldsymbol{S}\subseteq\boldsymbol{R}^n$ 是在 $\boldsymbol{R}^n$ 上的限制集合,$f:\boldsymbol{S}\rightarrow\boldsymbol{R}^n$ 是一个 n 维函数,问题是找到一个适应度最优的个体 $\boldsymbol{x}_{\min}\in\boldsymbol{S}$,使得 $\forall\boldsymbol{x}\in\boldsymbol{S}$:$f(\boldsymbol{x}_{\min})\leqslant f(\boldsymbol{x})$。基本的进化策略算法步骤如下:

(1) 初始化　由 n 个个体构成的种群,每个个体表示为一对实数向量 (x_i,σ_i),其中 x_i(即第 i 个个体)是一个可以连续取值的变量,σ_i 称为步长,是一个微小的变动量,其中,$\forall i\in\{1,2,\cdots,\mu\}$。设置迭代计数器 $g=0$;

(2) 变异　每一个父个体平均产生 λ/μ 个子个体,总共产生 λ 个子个体

$$\boldsymbol{x}_m^{(g+1)}=\boldsymbol{x}_k^{(g+1)}+\boldsymbol{z}^{(g\lambda+m)} \tag{1}$$

式(1)中,$\boldsymbol{z}^{(g\lambda+m)}$ 是由 σ_i 计算得到的正态分布随机向量,其中 $k=1,\cdots,\mu$,$m=1,\cdots,\lambda$;

(3) 选择　计算每个子个体的适应度,选择 μ 个适应度最优的子个体作为下一代的父个体。如满足停止条件,则终止优化过程,否则令 $g=g+1$,返回至步骤(2)。

在进化策略中,个体可以直接表示为实数变量,而遗传算法的个体则是由变量转化的二进制编码表示。由于进化策略强调在进化过程中父代到子代的自适应性和多样性,它可以应用于离散空间问题的求解,同时也不需要单调性假设的支持,因此在优化过程中不易陷入局部最优。

3　基于进化策略的运动估计算法

基于简单进化策略(adaptive evolution strategies motion estimation,AESME)的算法[5],其基本要素包括:个体的定义、适应度函数、步长的自适应控制、终止条件和选择方法,详细描述如下。

3.1　个体的定义

在块匹配算法中,块匹配问题的解为最小匹配误差,即相对于当前块的空间位移矢量 $\boldsymbol{d}_{x,y}$,由该矢量得到的个体的定义如下:

$$\boldsymbol{C}_i^{(g)}=(x_i^{(g)},y_i^{(g)},\sigma_{x_i}^{(g)},\sigma_{y_i}^{(g)}) \tag{2}$$

其中,g 表示进化代数;$x_i^{(g)},y_i^{(g)}$ 为进化 g 代后第 i 个个体的坐标,其分别对应于 $\boldsymbol{d}$ 的 x 和 y 分量;$\sigma_{x_i}^{(g)},\sigma_{y_i}^{(g)}$ 分别是 $x_i^{(g)}$ 和 $y_i^{(g)}$ 的步长控制参数。$\forall i\in\{1,\cdots,\lambda\}$,$\forall(x_i^{(g)},y_i^{(g)})\in\boldsymbol{\Omega}$,$\boldsymbol{\Omega}$ 为搜索窗口内的所有点的集合。

3.2　变异操作

在进化策略中,变异操作是产生新个体的最主要方法。个体中变量的变异操作相互独立,父变量加上一个期望值为 0,方差为 σ 的正态分布随机数即得到新的子变量。变异操作定义如下:

$$x_i^{(g+1)}=x_i^{(g)}+\sigma_{x_i}^{(g)}\cdot N(0,1) \tag{3}$$

$$\sigma_{x_i}^{(g+1)}=\sigma_{x_i}^{(g)}\cdot\exp[\hat{\tau}\cdot N(0,1)+\tau\cdot N_{x_i}(0,1)] \tag{4}$$

其中,$N(0,1)$ 表示一个期望值为 0,方差为 1 的正态分布随机数,$N_{x_i}(0,1)$ 表示对于每一个'x_i'都重新

生成随机数。$\hat{\tau}$ 和 τ 是算子集参数，分别表示变异时的整体步长和个体步长，通常设定的值分别为 $(\sqrt{2\sqrt{n}})^{-1}$ 和 $(\sqrt{2n})^{-1}$。另外，对 $y_i^{(g)}$ 和 $\sigma_{y_i}^{(g)}$ 也进行相同的变异操作。如果变异后的变量值超出了搜索范围，则用取模操作对其进行规格化。

变异后的子代与父代的基因型只有微小的差异，而进化策略则强调演化过程中搜索方向和步长的自适应调节。由式(3)和式(4)可以看出，如何调整步长对算法的收敛速度至关重要。

3.3 步长的自适应控制

为使优化过程运行更高效，必须不断地修改步长。如果步长太小，则会造成过多不必要的搜索运算；相反，如果步长太大，搜索过程则可能会接近最优点，但不能逐渐靠近最优点，或者搜索方向会远离最优点，进而陷入局部优化。在进化策略中，步长的控制与算法的收敛性能紧密相关。

最简单的步长自适应控制机制是 1/5 规则[4]，即定义成功率为变异成功(指变异后的个体适应度值优于父亲的适应度值)的次数与已完成的变异次数的比值，在优化过程中，如果成功率大于 1/5，则减小步长，如果成功率小于 1/5，则增大步长。在实际应用中，1/5 规则对于子代个体数比较小时，取得了接近最优进化速率的效果[6]。

但是在块匹配算法中，由于考虑到计算消耗，λ 不可能取很大的值，因此为进一步提高优化效率，必须增加步长调整的频率。本文定义新的 $1/\lambda$ 规则为：每 λ 次变异，如果成功率大于 $1/\lambda$，则减小步长，如果成功率小于 $1/\lambda$，则增大步长。对于较小的 λ 和较小的循环代数，$1/\lambda$ 规则取得了比 1/5 规则更加优化的结果。在第 5 节将给出相应的试验结果。

3.4 适应度函数的定义

在进化策略中，每个个体的适应度等于所对应的目标函数，通常对目标函数不进行任何变换。基于简化计算复杂度的考虑，个体的适应度函数定义如下：

$$f(\boldsymbol{C}_i^{(g)}) = SAD(x_i^{(g)}, y_i^{(g)}) = \sum_{\Omega}\sum \left| I_t(m,n) - I_{t-1}(m + x_i^{(g)}, n + y_i^{(g)}) \right| \tag{5}$$

其中，SAD 表示当前宏块与参考宏块像素差的绝对值，Ω 代表搜索空间，$I_t(m,n)$ 表示在第 t 帧中的点 (m,n) 的像素值。

3.5 选择操作的定义

在进化策略中，选择操作是按照一种确定的方式进行，有两种选择方法，一种是 $(\mu+\lambda)$-选择，另一种是 (μ,λ)-选择，前者是指从 μ 个父个体和 λ 个子个体中选择 μ 个最优个体作为下一代的父个体，后者是只从 λ 个子个体中选择 μ 个最优个体作为下一代的父个体。

由于 $(\mu+\lambda)$-选择会阻止适应度差的子个体，使一个父个体可能存活很多代，而一个成功变异的子个体仍然具有较差的适应步长，这样将使使优化过程易于陷入局部最优。

本文采用 (μ,λ)-选择，由于它易于跳出局部最优，同时由于采用 $1/\lambda$ 规则加快了步长的调整频率，从而保证了算法在有限进化代数下的收敛速率。这样在搜索过程中，即可存储每一代中适应度最优的个体

3.6 终止条件

一般可定义以下两个停止条件，若满足条件(1)或(2)，则停止搜索。

(1) 当前的最小适应度值小于或等于阈值 T，T 定义为前向帧的最小适应度值，其初始值为 0；

(2) 种群进化代数达到最大代数。

由于运动估计是基于 2 维平面中刚体的平移运动模型，因此在当前的算法中不考虑重组操作。通常设定 $\mu=1$，因为 $(1,\lambda)$-ES 算法只采用变异操作，结构简单，同时，由于 $(1,\lambda)$-ES 算法可以产生较大的步长搜索适应度值差别大的点[6]，所以当搜索的方向正确时，则在相应方向减小步长搜索。

综上所述，AESME 算法流程如下：

(1) 用运动向量(0,0)来初始化种群，设定 $g=0$，计算父亲的适应度，然后将 $\boldsymbol{C}_0^{(g)}$ 赋给父代适应度最优的个体 $\boldsymbol{C}_{\min}$，并在 $\boldsymbol{C}_{\min}$ 中存储当前适应度最优的个体；

$$\boldsymbol{P}^{(g)} = \{\boldsymbol{C}_0^{(g)}\}, \boldsymbol{C}_0^{(g)} = \{0,0,\sigma_{x_0}^{(g)}, \sigma_{y_0}^{(g)}\}$$

(2) 执行 λ 次变异操作，产生以下 λ 个子个体：

$$\boldsymbol{P}^{(g+1)} = \{\boldsymbol{C}_1^{(g+1)}, \boldsymbol{C}_2^{(g+1)}, \cdots, \boldsymbol{C}_\lambda^{(g+1)}\}$$

(3) 计算种群 $\boldsymbol{P}^{(g+1)}$ 中每个个体的适应度；

(4) 按个体适应度对种群 $\boldsymbol{P}^{(g+1)}$ 进行排序，将得到的适应度最优的个体 $\boldsymbol{C}_{\min}^{(g+1)}$ 作为下一代的父个体来得到新的种群，然后更新 $\boldsymbol{C}_{\min}$

$$\boldsymbol{P}^{(g+2)} = \{\boldsymbol{C}_{\min}^{(g+1)}\}$$

(5) 检测停止条件，如果满足，则转到步骤(7)；

(6) 按照第 3.3 节方法调整步长，令 $g=g+1$，

转至步骤(2);

(7) 由 $C_{\min}$ 得到最后的运动向量。

4 基于相关变异进化策略的运动估计算法

基于相关变异进化策略的自适应运动估计(adaptively correlated evolutionStrategies motion estimation,ACESME)算法是在 AESME 算法的基础上的改进算法,即在 ACESME 算法中引入了运动方向信息,以提高算法的进化速率和搜索的精确度,同时,采用种群规模的自适应控制和适应度计算的早期停止方法来进一步减小算法的复杂度。

4.1 个体定义

在 AESME 算法中,由于 x 和 y 变量的变异相互独立,各自沿坐标轴进行爬山式搜索,且通常情况下,宏块的运动方向并不沿着坐标轴,因此,只有当变异结果和宏块的运动方向吻合,即产生相关时,才能达到最优的进化速率。为了进一步提高进化速率,减少不必要的搜索步骤,可引入运动方向信息 θ,同 x,y 一样,θ 也是个体的一个变量,可进行类似的变异操作。这样搜索过程通过自学习过程就可以沿任一方向进行搜索。ACESME 算法的个体定义如下:

$$C_i^{(g)} = (x_i^{(g)}, y_i^{(g)}, \sigma_{x_i}^{(g)}, \sigma_{y_i}^{(g)}, \theta_i^{(g)}, \Delta\theta_i^{(g)}) \quad (6)$$

其中,$\Delta\theta_i^{(g)}$ 是变量 $\theta_i^{(g)}$ 的步长控制参数。

4.2 相关变异操作

在 ACESME 算法中,x,y 变量通过运动方向信息 θ 进行相关变异,变异操作定义如下:

$$\theta_i^{(g+1)} = \left(\theta_i^{(g)} + N_\theta\left(0, \left(\frac{5\pi}{180}\right)^2\right) + \pi\right) \bmod 2\pi - \pi \quad (7)$$

$$\Delta x_i^{(g+1)} = \sigma_{x_i}^{(g+1)} \cos(\theta_i^{(g+1)}) - \sigma_{y_i}^{(g+1)} \sin(\theta_i^{(g+1)}) \quad (8)$$

$$x_i^{(g+1)} = x_i^{(g)} + \Delta x_i^{(g+1)} \quad (9)$$

$\sigma_{x_i}^{(g)}$,$\sigma_{y_i}^{(g)}$ 变异同式(4)。对 $y_i^{(g)}$ 进行类似的变异操作即得到 $y_i^{(g+1)}$。

由于图像的整体运动和图像中较大物体的运动使相邻的宏块具有类似的运行方向,所以,当前宏块搜索得到的最优个体的运动信息变量可作为下一个相邻宏块搜索的初始值。

4.3 种群规模的自适应控制

在已经提出的基于遗传算法的运动估计方法和 AESME 算法中,种群大小在这个搜索过程中是固定的,而且在搜索过程中,存在一个最优的种群大小值[7],如果种群太小,则搜索过程易于陷入局部最优;相反,如果种群过大,则会增加算法的复杂度。为了在保证算法的收敛速率的前提下,减少算法的计算复杂度,本文引入种群规模的自适应机制。

$$\lambda^{(g+1)} = \lambda^{(g)} \exp\left(\beta_\lambda \frac{\Delta f_2}{\sqrt{\sum_{i=1}^{\lambda} (\Delta f_i)^2/(\lambda - 1)}}\right) \quad (10)$$

其中,$\lambda^{(g)}$ 表示第 g 代的子代种群个数,Δf_i 表示第 i 个最佳的适应度值与父亲的适应度的差值,子个体数量 λ 调整的强度 β_λ($\beta_\lambda \geqslant 0$)用于决定自适应调整的速率,$\beta_\lambda$ 的值越大,$\lambda^{(g)}$ 的随机浮动越大,如果 $\beta_\lambda = 0$,则种群规模保持不变。

这是一种确定性的自适应控制方法,其只需要很小的计算代价,即可在每代搜索中只产生优化个数的子个体,这就避免了不必要的计算消耗。

4.4 适应度计算的早期停止

在运动估计过程中,运算量消耗最大的是适应度计算,种群规模的自适应控制虽可以减少不必要的个体适应度计算,但在适应度计算过程中,如果累加值已经大于当前最优个体的适应度值,那么当前个体就不可能成为最优个体,此时适应度的计算应早期停止,以减少多余的计算。

在 ACESME 算法中,$\mu = 1$,采用(μ,λ)-选择算子。在每一代的进化循环中,只选择一个适应度最优的子个体作为下一代的父个体,并以种群中当前适应度最优的子个体的适应度值作为一个阈值,这样在每个子个体的适应度计算过程中,如果累加值超过阈值,则停止计算,同时舍弃当前子个体。

此外,在建立搜索点和适应度值的列表时,对于已计算过的搜索点,在进化循环过程中,可以直接返回其适应度值,以减少冗余计算。

在 ACESME 算法中,也设定 $\mu = 1$,且该算法的终止条件、步长的自适应控制、适应度定义及选择操作与 AESME 算法相同。

综上所述,ACESME 算法的流程如下:

(1) 用运动向量(0,0)初始化种群,设定 $g = 0$,计算父亲的适应度,并将 $C_0^{(g)}$ 赋给父代适应度最优的个体 $C_{\min}$;

$$P^{(g)} = \{C_0^{(g)}\}, C_0^{(g)} = \{0, 0, \sigma_{x_0}^{(g)}, \sigma_{y_0}^{(g)}, \theta_0^{(g)}, \Delta\theta_0^{(g)}\}$$

(2) 执行 $\lambda^{(g)}$ 次变异操作,产生 $\lambda^{(g)}$ 个子个体,即

$$P^{(g+1)} = \{C_1^{(g+1)}, C_2^{(g+1)}, \cdots, C_{\lambda^{(g)}}^{(g+1)}\}$$

(3) 计算种群 $P^{(g+1)}$ 中每个个体的适应度;

(4) 将得到的适应度最优的个体 $C_{\min}^{(g+1)}$ 作为下一代的父个体,再更新 $C_{\min}$,即得到新的种群。

$$P^{(g+2)} = \{C_{\min}^{(g+1)}\}$$

(5) 检测停止条件,如果满足,则转到步骤(8);

(6) 按照 4.3 节所述方法计算 $\lambda^{(g+1)}$;

(7) 按照第 3.3 节方法调整步长,使 $g = g + 1$,转至步骤(2);

(8) 由 $C_{\min}$ 得到最后的运动向量。

5 仿真结果和性能评价

为验证本文算法的性能,在 MoMusys MPEG-4 VM 平台上实现了基于进化策略的运动估计算法,并进行了算法的仿真和验证。算法实现了 SP@ L3 功能,并禁用错误恢复工具。为便于比较,宏块大小固定为 16 × 16,搜索窗口大小为 15 × 15,图像格式为 CIF,帧频为 30f/s,最大传送速率为 384bit/s,I 帧和 P 帧间隔为 1,该算法很容易被扩展到 B 帧。算法的参数,$\mu = 1$,$\hat{\tau} = 0$,$\tau = 0.7$;在 AESME 算法中设定 $\lambda = 8$;在 ACESME 算法中,$\beta_\lambda = 0.03$,$4 \leqslant \lambda \leqslant 8$;最大的搜索代数设定为 7。仿真测试时,采用以下 5 个序列的前 90 帧作为测试数据,即 News,Coastguard,Bus,Stefan,和 Flower 序列。

表 1 中所示为基于不同步长控制方法实现的 AESME 算法的性能比较,包括没有步长控制、1/5 规则和 1/λ 规则 3 种情况。采用编码重建图像的峰值信噪比(peak signal-to-noise ratio,简称 PSNR)值作为性能对比参数。从表 1 中可以看出,1/λ 规则的图像质量最好,与 1/5 规则相比,平均提高了 0.117dB。对于包含快速运动的图像序列,如 Bus,1/λ规则的峰值信噪比值较其他两种方法性能提高比不包含快速运动的序列大,这主要因为 1/λ 规则加快了步长自适应变化的频率,所以对于最大循环代数较小的情况,可提高搜索的收敛速率。

表 1 基于 AESME 算法的 1/5 规则和 1/λ 规则性能(PSNR)比较 (单位:dB)

Tab.1 Performance comparison of 1/5 rule and 1/λ rule based AESME algorithm(PSNR)

序列	算法		
	无	1/5 规则	1/λ 规则
News	39.878	40.080	40.091
Coastguard	36.044	36.067	36.190
Bus	35.800	35.813	36.095
Stefan	35.073	35.122	35.153
Flower	36.201	36.223	36.360
Average	36.599	36.661	36.778

表 2 为各种运动估计方法的性能(即编码重建图像的峰值信噪比)比较。从表 2 可以看出,ACESME 算法的性能与全搜索(full search,FS)算法接近,好于三步法(three step search,TSS)和 AESME 算法。同 AESME 算法相比,ACESME 算法编码图像的峰值信噪比平均提高了 0.5dB,对于包含快速运动的序列,如 Bus,Stefan,其性能提高比其他序列大。由于 ACESME 算法引入了运动方向变量,因而使 x 和 y 分量的变异不再是互相独立,而是相关的,这样不仅使搜索过程能更快地向最优的运动方向靠近,而且使搜索的点分布更合理,也提高了算法的收敛速率和精确度。同时,每个变量都有一个相应的步长参数,从而使算法更具灵活性,相反,三步法每一步只对固定的点进行搜索,缺乏灵活性。

表 2 算法的性能(PSNR)比较 (单位:dB)

Tab.2 Performance Comparison of algorithms

序列	算法			
	FS	AESME	ACESME	TSS
News	40.123	40.091	40.113	40.002
Coastguard	37.119	36.190	36.910	34.652
Bus	37.174	36.095	36.870	34.463
Stefan	37.958	35.153	35.898	32.791
Flower	37.702	36.360	36.681	35.020
Average	38.015	36.778	37.294	35.386
百分比(%)	100	96.7	98.1	93.1

为比较 3 算法的运算量,对上述的 5 个序列进行了算法运算时间测试,测试时,每种算法运行 10 次,计算每种算法的平均执行时间。这里只累加运动估计模块的执行时间,时间单位精确到 ms。图 1 为 3 种算法的平均执行时间比较图,由该图可以看出,AESME 和 ACESME 算法的运算量远远低于全搜索算法,仅比三步法略高。因为 AESME 和 ACESME 算法除 SAD 操作消耗外,还要进行变异和选择操作,故时间略长。

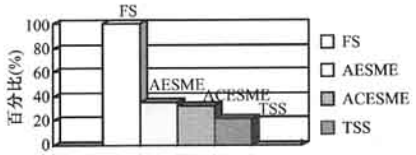

图 1 算法平均执行时间比较图

Fig. 1 Average executive time comparison chart of algorithms

由于采用了种群大小的自适应控制,ACESME 算法所产生的全部子个体数减少了大约 10%,而采

用适应度计算早期停止方法又使 SAD 累加次数减少约 5%,虽然 ACESME 执行相关变异操作增加了算法的复杂度,但在图 1 中却显示 ACESME 算法和 AESME 算法的复杂度相差很小。

6 结　论

本文提出了一种新的快速运动估计方法——ACESME 算法。这种 ACESME 算法将进化策略应用到运动估计的搜索过程中,并通过符合正态分布的变异操作和有效的步长自适应控制来取得较好的收敛速度,同时第 1 次将运动信息变量引入进化过程,从而进一步提高了算法的收敛速率和搜索的精确度。另外,为降低算法的计算复杂度,还采用了种群规模的自适应控制和适应度计算的早期停止方法,ACESME 算法虽然采用复杂度较高的相关变异操作,但总体计算复杂度与 AESME 方法类似。实验结果表明,ACESME 算法的编码性能与全搜索算法相近,而编码的复杂度仅略高于三步法。由于 ACESME 算法中不包含重组操作,因此算法的结构比其他基于遗传算法的运动估计方法更简单,同时由于进化策略的内在并行性,从而使该算法适合于硬件实现。

参考文献(References)

1 Lin Chun-Hung, Wu Ja-Ling. A lightweight genetic block-matching algorithm for video coding[J]. IEEE Transactions, Circuits System. Video Technology, 1998,**8**(4): 386 ~ 392.

2 Li Shen, Xu Wei-Pu, *et al.* A novel fast motion estimation method based on genetic algorithm[A]. In: Proceeding of IEEE International Conference on Image Processing[C]. Kobe Japan, 1999,**1**: 66 ~ 69.

3 Xu YueLei, Bi Duyan, Mao Baixin. A genetic search algorithm for motion estimation[A]. In: Proceeding of International Conference on Signal Processing[C], BeiJing, China, 2000,**2**: 1058 ~ 1061.

4 Schwefel H P. Evolution and optimum seeking[M]. New York: Wiley&Sons,1995.

5 Wang Hui, Mao Zhi-Gang. An adaptive motion estimation algorithm based on evolution strategies [A]. International Conference on Acoustics, Speech, and Signal Processing[C], Montreal Canada, 2004,**3**: 353 ~ 356.

6 Beyer H G. Toward a theory of evolution strategies: Self-adaptation [J]. Evolutionary Computation, 1996,**3**(3): 311 ~ 347.

7 Nikolaus Hansen, *et al.* Sizing the population with respect to the local progress in $(1,\lambda)$-evolution strategies—A theoretical analysis[A]. In: IEEE International Conference on Evolutionary Computation Proceedings[C], Perth, Australia, 1995:80 ~ 85.