AN ADAPTIVE MOTION ESTIMATION ALGORITHM BASED ON EVOLUTION STRATEGIES

Wang Hui, Mao Zhigang

Harbin Institute of Technology Microelectronics Center

ABSTRACT

Based on evolution strategies (ESs) a novel adaptive motion estimation search algorithm (AESME) is presented. ESs consider the evolution progress on the phenotype level. In contrast, genetic algorithms focus on heredity genetic mechanism on the chromosomes level. In ESs the mutation operation accords with the normal distribution law. In the AESME algorithm the (μ, λ)-ES algorithm is adopted to block motion estimation, and the adaptive scheme is advanced to improve the convergence rate on the basis of the 1/5 success rule. Experimental results demonstrate that this algorithm has similar performance to that of the full-search (FS) algorithm, and owing to the inherent parallelism and low complexity of ESs, AESME is suitable for VLSI implementation.

1. INTRODUCTION

In the video coding system, such as H.264, MPEG-4 and AVC, the motion estimation is an essential component of the encoding algorithm. For it is the computationally most demanding algorithm of an encoder, taking up approximately 60-80% of the total computation time, the motion estimation algorithm has a high impact on the visual performance of an encoder at a given bit rate.

The common motion estimation method is the block-matching algorithms (BMAs). Among the BMAs, the FS method leads the best result. However, the FS's high computational cost prevents it from being applied in most real-time systems. A number of fast algorithms have been proposed, such as three-step search, diamond-search and some predictive search algorithms. These algorithms are mainly based on the assumption, i.e. the matching function changes monotonously when the search point varies from the farthest point to the optimal one, and this assumption are not always accord with the real cases. Moreover in these algorithms, during each search, only fixed points are checked. So they are inclined to the local optima.

In order to improve the performance degraded by the monotonous assumption, genetic algorithms (GAs) have been applied to BMA, in [1] [2] [3], the search algorithms based on genetic algorithms are proposed. But the features of GAs needs sufficient individuals and generations to get the optimal result, in [1] – [3], the population size adopted is equal to or larger than 16, which makes it ungainly to decrease the computational complexity.

Though both GAs and ESs are the major branches of evolutionary algorithm, they are closely related but respectively developed. The former focus on heredity genetic mechanism on the chromosomes level, while the latter consider the evolution progress on the phenotype level, in which only selection and mutation are often used to solve problems. In addition, the mutation operation accords with the normal distribution law in ESs. ESs are significantly faster than traditional GAs in numerical optimization and also more likely to find true global extremum of a function [4].

In this paper, an adaptive motion estimation method based on ESs is provided. In Section 2, the basic principles of ESs are introduced. In Section 3, the AESME algorithm is described. In Section 4 the corresponding experimental results as well as performance comparisons with other algorithms will be presented.

2. THE BASIC PRINCIPLES OF ES

ESs were introduced in 1964 by Rechenberg working in Berlin and further developed by Schwefel [4]. ESs were initially designed to solve difficult discrete and continuous parameter optimization problems. Although recombination has been used in ESs, the mutation is the primary search operator.

A principal one of ESs is (μ, λ)-ES, where $\lambda \geq \mu \geq 1$. (μ, λ) means that μ parents generate λ offspring in each generation. In this paper, only the simple version of ESs, i.e. ESs without any recombination, is considered. A global minimization problem can be formalized as a pair (*S, f*), where $S \subseteq R^n$ is a bounded set on R^n and $f: S \rightarrow R^n$ is an n-dimensional function, the problem is to find an $x_{\min} \in S$, so that $\forall x \in S: f(x_{\min}) \leq f(x)$. The basic algorithm is described as follows:

Step 0: (Initialization) A given population consists of μ individuals. Each individual is taken as a pair of realvalued vectors, (x_i, σ_i), x_i is the variable, and σ_i can be called as the step length. $\forall i \in \{1,2\ldots\mu\}$. Set the generation counter g=0;

Step 1: (Variation) Each individual parent produces λ/μ offspring on average, so that a total of λ offspring is generated. For k=1… μ and m=1…λ.

$$x_m^{(g+1)} = x_k^{(g+1)} + z^{(g\lambda+m)} \quad (1)$$

Where $z^{(g\lambda+m)}$ is a normally distributed random vector computed from σ_i;

Step 2: (Selection) Evaluate the fitness for each offspring, then only the μ best of the λ offspring become parents of the following generation. Stop if the stopping criterions are satisfied, otherwise g=g+1, and go to Step 1.

In the ESs, the individual can be directly expressed by the variables, however, in the GAs it is coded into binary representation of the variables. ESs are neither confined to continuous space, nor supported by the assumption of monotony. The qualities of ESs enable it not to be trapped to local optima. In the next section, the AESME block matching algorithm will be described in detail.

3. AESME SEARCH ALGORITHM

The basic elements in ESs are the definition of individuals, the control of parameters, fitness evaluating function, the terminating rules and selection method. The AMES search algorithm is discussed as follows.

3.1. The definition of the individuals

In BMA, the motion vector is represented by $\overrightarrow{MV}$ (x, y), and the individual is defined as

$$C_i^g = (x_i^g, y_i^g, \sigma_{\mathrm{xi}}{}^{\mathrm{g}}, \sigma_{\mathrm{yi}}{}^{\mathrm{g}}) \quad (2)$$

where g is the generation number, $x_{\mathrm{i}}^{\mathrm{g}}$, $y_{\mathrm{i}}^{\mathrm{g}}$ correspond to the x and y variables of $\overrightarrow{MV}$. And $\sigma_{\mathrm{xi}}{}^{\mathrm{g}}$, $\sigma_{\mathrm{yi}}{}^{\mathrm{g}}$ is the step length of $x_{\mathrm{i}}^{\mathrm{g}}$ and $y_{\mathrm{i}}^{\mathrm{g}}$ respectively, $\forall i \in \{1,\ldots,\lambda\}$, $\forall (x_i^g, y_i^g) \in \Omega$, Ω represents the set of all points in the range of search window.

3.2. The mutation of the individuals

In ESs, mutation is performed independently on each other vector element by adding a normally distributed random value with expectation zero and standard deviation σ, the operation of mutation is defined as:

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

$$\sigma_{xi}^{g+1} = \sigma_{xi}^g \cdot \exp[\tau' \cdot N(0,1) + \tau \cdot N_{xi}(0,1)] \quad (4)$$

where $N(0,1)$ denotes a normally distributed one-dimensional random number with mean zero and standard deviation one, and $N_{xi}(0,1)$ indicates that the random number is generated anew for each value of 'xi', the factors τ' and τ usually set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$. The mutation of y_i^g and $\sigma_{yi}{}^g$ is identical. If the coordinates after mutation exceed the search range, then adjust the coordinate's variables by applying the modulus operation.

The genotype of a descendant differs only slightly from that of its parents. From Eq. (3) and Eq. (4), it can be seen that how to control the step length parameter σ is very important to the convergence rate.

3.3. Step length adaptive control

The simplest step length adaptive control is conducted according to the 1/5 rule, that is, during the optimum search the frequency of successes is obtained, i.e. the ratio of the number of successes to the total number of trials (mutations), if the ratio is greater than 1/5, increase the variance, if it is less than 1/5, decrease the variance [4]. This rule simply adopts the mutation strength through a deterministic and rigid control mechanism, when a relative small λ is adopted, this rule obtains the ideal result.

But in the BMA, considering the computation cost, the λ and the maximum generation could not adopt too large numbers. In order to improve the convergence rate further, the step length adaptive control, the 1/λ rule, is defined as: during the optimum search, every 3λ mutations, if the ratio is greater than 1/λ, increase the variance, if it is less than 1/λ, decrease the variance. For the small λ (λ<5) and maximum generation number, it obtains the better result than the 1/5 rule. This will be shown in Section 4.

3.4. The definition of fitness function

The fitness function is defined as:

$$f(C_i^g) = SAD(x_i^g, y_i^g) = \sum_{\Omega}\sum \left| I_p(m,n) - I_{p-1}(m + x_i^g, n + y_i^g) \right| \quad (5)$$

where SAD represents the sum of absolute differences, Ω represents the search space, $I_p(m,n)$ means pixel value of the point (m, n) in the frame p.

3.5. Selecting method

There are two main selection methods in ESs, namely, (μ, λ)-selection and $(\mu+\lambda)$-selection. In the former method, the parents of the next generation are obtained by selecting the μ best offspring. While in the latter one, the parents are obtained from both the older generation and the offspring.

Because the $(\mu+\lambda)$-selection permits no worsening of the fitness value, the parent survives and may do so for many generations, which increases the probability of a successful mutation still having a poorly adapted step length. If a small λ and small maximum generation number are adopted, (μ+λ)-selection will remarkably decrease the convergence rate and the algorithm is prone to trapping into local optima.

In the AESME algorithm, the (μ, λ)-selection is applied, and the individual with minimum fitness value is stored for each every generation.

3.6. The definition of terminating rules

There are two stopping rules defined, either condition a) or b) is satisfied then the search stops.

a) The current minimum fitness is less than or equal to the threshold value *TH*. *TH* is assigned to the minimum value of fitness in the previous frame, and *TH* is initialized as zero.

b) The generation number reaches the maximum generation number.

In the current implementation of AESME algorithm, set μ=1. Because the structure of (1, λ)-ES is simpler, and (1, λ)-ES can perform large search steps with the result of larger fitness difference, (1, λ)-ES makes only a reduced size stop in that direction[5] when it found the right direction. The AESME algorithm starts from the small step length, then adaptively adjusts the step length according to the 1/λ rule. The AESME algorithm can be summarized as follows:

Step 0: initialize the population with the motion vector (0,0), g=0, calculate the fitness of the parent, assign C_0^g to $C_{\min}$;

$$P^g = \{C_0^g\}\ ,\ C_0^g = \{0,0,\sigma_{x0},\sigma_{y0}\}$$

Step 1: execute λ times mutation operations to generate λ offspring;

$$P^{g+1} = \left\{C_1^{g+1}, C_2^{g+1}, ..., C_\lambda^{g+1}\right\}$$

Step 2: evaluate the fitness of each individual in P^{g+1};

Step 3: sort the P^{g+1} according to individuals fitness, get the $C_{\min}{}^{g+1}$, update the $C_{\min}$, obtain the new population;

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

Step 4: stop the search if the stopping rules are satisfied, go to Step 6;

Step 5: adjust the step length according to the section 3.3, g=g+1, go to Step 1;

Step 6: obtain the $\overrightarrow{MV}$ from $C_{\min}$.

4. SIMULATION RESULTS AND DISCUSSION

In the simulations, AESME is compare with FS, TSS. AESME algorithm was completed in MoMuSys MPEG4 VM platform, using the simple profile, error resilience mode is disabled. The block size is fixed at 16x16. To make a consistent comparison, block matching is conducted within a 15x15 search window, picture format is CIF, frame rate is 30f/s, and the distance between *P* frames is 1, it is easy to extend to *B* frames. As for the parameters of ESs, the λ = 3, the τ'=0, τ=0.7, the maximum generation number is 9. We select first 75 frames of the sequences *Bus, Foreman, News, Mobile,* and *Garden* to test the proposed algorithm.

Table 1
Comparisons of PSNR of 1/5-rule and 1/λ rule

adaptive control Image(CIF)	no control	1/5 rule	1/λ rule
Bus	36.139	36.153	36.326
Foreman	37.461	37.470	37.503
News	39.879	40.035	40.036
Mobile	35.071	35.094	35.213
Garden	36.690	36.710	36.848
Average	37.048	37.092	37.185

A performance comparison among the three algorithms is shown in Table 1, based on ESs without the adaptive control, with 1/5 rule and with 1/λ rule in terms of their average peak signal-to-noise ratio (PSNR) after reconstruction. It can be seen that, of all the sequences, the 1/λ rule is the best, and it improves 0.09dB on average compared with to the 1/5 rule. The image sequences with fast motion, like *Bus* and *Mobile*, the quality improved is larger than the other sequence with low motion, because the 1/λ rule provide higher frequency step length adjustment than the 1/5 rule, which helps increase the convergence rate. However, the quality of the algorithm with the 1/λ rule does not increase largely in contrast to the one without any adaptive control. A possible explanation is that the *x* and *y* variables are mutated independently, if the direction information is added to individual, the quality will be further enhanced.

Table 2
Comparisons of PSNR of the Search Algorithms

ME method Image	FS	TSS	AESME
Bus	37.159	34.689	36.326
Foreman	38.355	34.306	37.503
News	40.093	39.972	40.036
Mobile	36.009	33.930	35.213
Garden	37.651	34.973	36.848
Average	37.853	35.574	37.185
Percentage	100%	94%	98%

The difference in PSNR of FS, TSS and AESME algorithm is shown in Table 2. From Table 2, it can be observed that the performance of AESME is similar to that of FS, and much better than that of TSS. For the AESME algorithm introduces the good step length generation and adaptive control mechanism, which makes the searched points distribute more reasonably, and each variable with a step length parameter endows the algorithm with more flexibility. Whereas, TSS just searches fixed points without much flexibility.

In order to compare the computational complexity of three algorithms, each algorithm is executed ten times for every sequence to get the average execution time. The execution time is accumulated only for motion estimation module; it is expressed in the unit of microsecond. The rate graph is shown in Fig.1. The execution time of AESME is far less than that of FS, but a little more than that of TSS. Because AESME needs to execute mutation operations λ times in each generation besides the *SAD* operations, generally, AESME requires 27 (λ * maximum generation number) points, while TSS only search fixed number of points in each macro block.

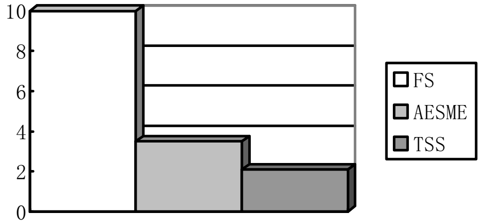

Fig.1. Comparison of the Average Execution Time

5. CONCLUSION

This paper proposed a new fast search algorithm for block matching motion estimation. AESME algorithm is derived from ESs, by means of mutation operation according to the normal distribution law, and obtains good convergence rate. The experiment results show that the performance of AESME is similar to that of FS, and the computational complexity is just a little larger than that of TSS. The AESME algorithm has less kinds of operations and much simpler structure than other motion estimation algorithms based on GAs, AESME algorithm does not include recombination operation, and due to the inherent parallelism of ESs, AESME algorithm is suitable to the VLSI implementation.

6. THE FUTURE RESEARCH

In order to improve the efficiency and reduce the complexity of AESME algorithm, our future research mainly focuses on three aspects. First, in the current version of AESME, the *x* and *y* are mutated independently, there is no direction information in the definition of the individuals. Some factors of direction information should be introduced into our algorithm. Second, to reduce the computation cost in mutation operation, some other simpler mutation methods is considered to use, such as the mutation rule given by the symmetrical two-point distribution. Finally, through using the predictive algorithm to select the initial parent will improve the convergence rate further.

7. REFERENCES

[1] Chun-Hung Lin, Ja-Ling Wu, “A Lightweight Genetic Block-Matching algorithm for Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol.8, pp.386-392, Aug, 1998.

[2] Shen Li, Wei-Pu Xu et al, “A novel fast motion estimation method based on genetic algorithm,” Proceeding of ICIP99, vol.1 pp.66-69, 1999.

[3] Xu YueLei, Bi Duyan, Mao Baixin, “A Genetic Search Algorithm for Motion Estimation,” Proceeding of ICSP2000, vol.2, pp.1058-1061, Aug, 2000.

[4] Schwefel H.-P, “Evolution and Optimum Seeking,” Wiley&Sons, NY, 1995.

[5] H.-G. Beyer, “Mutate Large, But Inherit Small! On the Analysis of Rescaled Mutations in ($\tilde{1},\tilde{\lambda}$)-ES with Noisy Fitness,” Parallel Problem Solving from Nature, 5, pp.109-118, Heidelberg, Springer, 1998.