AN ADAPTIVE MOTION ESTIMATION ALGORITHM BASED ON EVOLUTION STRATEGIES WITH CORRELATED MUTATIONS

Wang Hui, Mao Zhigang

Harbin Institute of Technology Microelectronics Center

ABSTRACT

Based on evolution strategies (ESs) with correlated mutations, a novel algorithm - adaptively correlated ES motion estimation (ACESME) is presented. ESs consider the evolution progress on the phenotype level. In contrast, genetic algorithms focus on heredity genetic mechanism on the chromosomes level. The mutation operation in ESs accords with the normal distribution law. In the ACESME algorithm, the (μ, λ)-ES algorithm with correlated mutations is adopted to block motion estimation. In this algorithm 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 (FS) algorithm. Furthermore, owing to the inherent parallelism and low complexity of ESs, ACESME is applicable for VLSI implementation.

1. INTRODUCTION

In the video coding system, such as H.264, MPEG-4 and AVC, the motion estimation is computationally the most demanding part of an encoder. It takes up approximately 60-80% of the total computation consumption. 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 to the best result. However, the FS's high computational cost prevents it from being applied in most real-time systems. Many fast algorithms have been proposed, such as three-step search, diamond-search and some predictive search algorithms. To describe the condition of a point in a two-dimension plane, two factors are needed, i.e. coordinate and motion direction factor. In most of these fast algorithms, the motion direction factor is implicit. Though it has been introduced by Jae-Yeal Nam et al [1] into their APDSA, it is just used to predict the best motion vector (MV) candidate, and as the direction factor is from the MVs of previous frame, additional memory is needed.

These fast algorithms are mainly based on the assumption, that the matching function changes monotonously when the search point varies from the farthest point to the optimal one. The real cases do not always accord with this assumption. 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 in BMA. In [2] [3], the search algorithms based on genetic algorithms are proposed. But the features of GAs need sufficient individuals and generations to get the optimal result, in [2] [3], the population size adopted is equal to or larger than 16, which makes it ungainly to decrease the computational complexity.

In this paper, an adaptive motion estimation method based on ESs with correlated mutations (CORR-ES) is provided. In Section 2, the principles of CORR-ES are introduced. In Section 3, the ACESME algorithm is described. In Section 4 the corresponding experimental results as well as comparisons of performance and complexity with other algorithms will be presented.

2. THE PRINCIPLES OF CORR-ES

Both GAs and ESs are the major branches of evolutionary algorithms. 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. The mutation operation in ESs accords with the normal distribution law. In addition, ESs are significantly faster than traditional GAs in numerical optimization and also easier to find true global extremum of a function [5].

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. Based on ESs, CORR-ES was proposed in 1981[6], and has become a standard algorithm [7], which is especially promising for difficult problem instances.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

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. Although recombination has been used in ESs, the mutation is the primary search operator. In this paper, CORR-ES 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 $f: S \to R^n$ is an n-dimensional function, the problem is to find an $x_{\min} \in S$, which makes $\forall x \in S: f(x_{\min}) \leq f(x)$. The basic algorithm is described as follows:

Step 0: (Initialization) A given population consists of μ individuals. Each individual is taken as $(x^g, \sigma^g_{i=1,\ldots,n}, \theta^g_{j=1,\ldots,n(n-1)/2}, \Delta\theta^g_{j=1,\ldots,n})$, x is the variable, and σ_i can be called as the step length of variable x, θ_j is the inclination angle vector, $-\pi \leq \theta_j < \pi$, $\Delta\theta_j$ is the angle step length. 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.

$$x^{g+1} = x^g + Z(\theta_1^{g+1}, \ldots \theta_{n(n-1)/2}^{g+1}) \bullet \begin{pmatrix} \sigma_1^{g+1} & & \\ & \ddots & \\ & & \sigma_n^{g+1} \end{pmatrix} \bullet N(0, I) \quad (1)$$

The result is rotated in all two-dimensional subspaces spanned by canonical unit vectors, denoted by $Z(\bullet)$. Where σ_i^{g+1} and θ_j^{g+1} are normally distributed random vectors computed from σ_i^g and $\Delta\theta_j^g$ separately, $N(0,I)$ is normally distributed random vector;

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 CORR-ES, the angle vector θ_j, rather than σ_i, represents the additional strategy variable. It is modified in a similar way like variable x. In this way, the CORR-ES may adapt to any preferred direction of search by means of self-learning. CORR-ES is not supported by the assumption of monotony. It can be applied in discrete space as well as in continues space. The qualities of CORR-ES enable it not to be trapped to local optima. In the next section, the ACESME block matching algorithm will be described in detail.

3. ACESME SEARCH ALGORITHM

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

3.1. The definition of the individuals

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

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

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 $\sigma_{\mathrm{xi}}^{\mathrm{g}}$, $\sigma_{\mathrm{yi}}^{\mathrm{g}}$ and $\Delta\theta_i^g$ are the step length of $x_{\mathrm{i}}^{\mathrm{g}}$, $y_{\mathrm{i}}^{\mathrm{g}}$ and θ_i^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 common ESs, each vector element mutates independently, but in CORR-ES they are correlated through angle variable. The operation of mutation is defined as:

$$\sigma_{xi}^{g+1} = \sigma_{xi}^g \cdot \exp[\tau' \cdot N(0,1) + \tau \cdot N_{xi}(0,1)] \quad (3)$$

$$\theta_i^{g+1} = (\theta_i^g + N_\theta(0, (5\pi/180)^2) + \pi) \bmod 2\pi - \pi \quad (4)$$

$$\Delta x_i^{g+1} = \sigma_{xi}^{g+1} \cos(\theta_i^{g+1}) - \sigma_{yi}^{g+1} \sin(\theta_i^{g+1}) \quad (5)$$

$$x_i^{g+1} = x_i^g + \Delta x_i^{g+1} N(0, I) \quad (6)$$

where $N(0,1)$ denotes a normally distributed one-dimensional random number with zero mean 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 σ_{yi}^g is same with that of σ_{xi}^g. $N_\theta(0, (5\pi/180)^2)$ denotes a normal distribution with zero mean and variance $(5\pi/180)^2$. Δx_i^{g+1} represents the change in the object variable x_i^g. The mutation of y_i^{g+1} is same with that of x_i^{g+1}. If the coordinates after mutation exceed the search range, then adjust the coordinate's variables through applying the modulus operation.

For the movement of the whole picture and large objects will produce similar motion vectors of adjacent macroblocks, the angle variable of last search will be the initial value of current angle variable.

From Eq. (3) and Eq. (5), it can be seen that how to control the step length parameter σ is very important to the convergence rate.

3.3. σ-adaptation scheme

The simplest step length adaptive control is conducted according to the 1/5 rule [5]. This rule simply adopts the mutation strength through a deterministic and rigid control mechanism. When a relatively small λ is adopted, this rule obtains the ideal result.

But in the BMA, considering the computational cost, the λ and the maximum generation could not adopt too large numbers. In order to improve the progress rate further, the step length adaptive control rule, is tuned as: during the optimum search, in each generation, if the

success ratio is greater than 1/λ, increase the variance; if it is less than 1/λ, decrease the variance. The success ratio defines as the ratio of the number of descendents whose fitness is better than the parent to λ. Better results can be obtained where λ and maximum generation number are small than under 1/5 rule.

3.4. The definition of fitness function

The fitness function is defined as:

$$f(C_i^g) = SAD(x_i^g, y_i^g) = \sum_{\Omega}\sum \left| I_p(m,n) - I_{p-1}(m + x_i^g, n + y_i^g) \right| \quad (7)$$

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

3.5. λ-adaptation scheme

In most application of ES, the λ is held fixed throughout optimization. Its value can be chosen so that the progress rate is optimal [7]. The λ-adaptation scheme used in ACESME is defined as:

$$\lambda^{g+1} = \lambda^g \exp(\beta_\lambda \frac{\Delta f_2}{\sqrt{\sum_{i=1}^{\lambda}(\Delta f_i)^2/(\lambda-1)}}) \quad (8)$$

where β_λ ($\beta_\lambda \geq 0$) determines adaptation speed, large values lead to fast adaptation, if it equals zero then no adaptation takes place; Δf_i represents the fitness difference between the i^{th} fittest offspring and parent.

This is a deterministic adaptation scheme for λ. It only needs a small computation expense. With a given maximum generation number, if a too small λ is adopted, the search is apt to local minima, whereas if λ is too big, the computation consumption will grow significantly. Through λ-adaptation, the computation consumption can be cut down without decreasing the progress rate.

3.6. Selecting method

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

Because the (μ+λ)-selection permits no worsening of the fitness value, which increases the probability of a successful mutation still having a poorly adapted step length. If a small λ and a small maximum generation number are adopted, (μ+λ)-selection will remarkably decrease the convergence rate and the algorithm is inclined to local optima.

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

3.7. The definition of terminating rules

There are two stopping rules:

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.

Either condition a) or b) is satisfied then the search stops.

In the current implementation of ACESME algorithm, set μ=1. Because the structure of (1, λ)-ES is simple and efficient, (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 when it finds the right direction. The ACESME algorithm starts from the small step length, and then adaptively adjusts the step length and the value of λ according to σ-adaptation and λ-adaptation scheme. The ACESME algorithm can be summarized as follows:

step1: initialize the population with the motion vector (0,0), set g=0, calculate the fitness of the parent, assign C_0^g to $C_{\min}$;

$$P^g = \{C_0^g\},\ C_0^g = \{0,0,\sigma_{x0}^g, \sigma_{y0}^g, \theta_0^g, \Delta\theta_0^g\}$$

step2: execute λ^g times mutation operations to generate λ^g offspring;

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

step3: evaluate the fitness of each individual in P^{g+1};

step4: sort the P^{g+1} according to individuals fitness, get the $C_{\min}^{g+1}$, update the $C_{\min}$, obtain the new population;

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

step5: stop the search if the stopping rules are satisfied, go to Step 8;

step6: compuate the λ^{g+1} according to section 3.5;

step7: adjust the step length according to the section 3.3, set g=g+1, go to Step 2;

step8: obtain the $\overrightarrow{MV}$ from $C_{\min}$.

4. SIMULATION RESULTS AND DISCUSSION

In the simulations, ACESME is compared with FS, TSS, and AESME[4]. ACESME algorithm has been completed in MoMuSys MPEG4 VM platform. The simple profile is used, and error resilience mode is disabled. The block size is fixed at 16x16, the maximum bitrate is 384kbit/s. 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, which is easy to extend to *B* frames. As for the parameters of ESs, the β_λ =0.03, $4 \leq \lambda \leq 8$, the τ'=0,

τ=0.7, the maximum generation number is 7. We select the first 90 frames of the sequences: *News, Coastguard, Bus, Stefan,* and *FlowerGarden* to test the proposed algorithm.

Table 1. Comparisons of PSNR of different algorithms

CIF \ methods	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
Percentage	100%	96.7%	98.1%	93.1%

A performance comparison among the four algorithms in terms of their average peak signal-to-noise ratio (PSNR) after reconstruction is shown in Table 1. It can be observed that the performance of ACESME is similar to that of FS, and much better than that of TSS. For the ACESME algorithm introduces the angle factor, which makes the mutations of *x*-variable and *y*-variable not independent but correlated, and in this way the progress rate and accuracy of search algorithm are enhanced. In addition, good step length generating method and adaptive control mechanism make the searched points distributing more reasonably. And each variable with a step length parameter endows the algorithm with more flexibility. On the other hand, TSS just searches fixed points without much flexibility.

It can be seen that the ACESME is better than AESME, and improves 0.5dB on average. In the image sequences with fast motion, such as *Bus* and *Stefan*, the quality improved is larger than the sequences with low motion, because using the angle variable makes the search more efficient and accurate. However, the quality of the ACESME does not increase largely in contrast to AESME. A possible explanation is that the λ-adaptation is adopted in ACESME, that is, the AESME searches more points.

In order to compare the computational complexity of four algorithms, each algorithm is executed ten times for every sequence to get the total number of offspring and the average execution time. The total number of offspring of ACESME is about 11% less than that of AESME. Only the execution time of motion estimation module is accumulated and expressed in the unit microsecond. The rate graph is shown in Fig.1. The execution time of ACESME is far less than that of FS, but a little more than that of TSS. Because ACESME needs to execute mutation operation λ times and other adaptation operations in each generation besides the *SAD* operations, while TSS only search fixed number of points in each macro block.

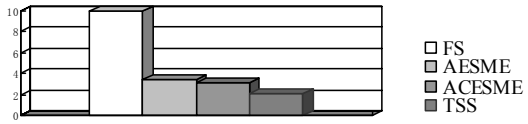

Fig.1. Comparison of the Average Execution Time

5. CONCLUSION

This paper has proposed a new fast search algorithm for block matching motion estimation. ACESME algorithm is derived from ESs. By means of correlated mutation operation, the angle variable is explicitly used in searching process, which enhances the progress rate and search accuracy. Furthermore, the population sizing adaptation scheme is used to decrease the computation complexity. The experimental results show that the performance of ACESME is similar to that of FS, and the computational complexity is just slightly larger than that of TSS. The ACESME algorithm has fewer kinds of operations and much simpler structure than other motion estimation algorithms based on GAs, and due to the inherent parallelism of ESs, ACESME algorithm is applicable for the VLSI implementation. In the future work, the VLSI implementation based on 32-bit embedded core SOC platform will be conducted.

6. REFERENCES

[1] Jae-Yeal Nam et al, "New fast-search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector," IEEE Trans. On Consumer Electronics, vol.46, No.4, pp934-942, Nov, 2000.

[2] 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.

[3] 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.

[4] Wang Hui, Mao Zhigang, "An adaptive motion estimation algorithm based on evolution strategies", Proceeding of ICASSP2004, Montreal, 2004.

[5] Schwefel H.-P, "Evolution and Optimum Seeking," Wiley&Sons, NY, 1995.

[6] Schwefel H.-P, "Numerical Optimization of Computer Models," Wiley, Chichester, 1981

[7] Nikolaus Hansen et al, "Sizing the population with respect to the local progress in (1, λ)-evolution strategies– A theoretical analysis," IEEE Conf. ICECP, pp80-85, 1995