

Evolutionary strategy search algorithm for fast block motion estimation

J. Hay and K.K. Loo

The evolutionary strategy search (ESS) algorithm is a novel method for implementing fast block motion estimation (ME) using evolutionary strategy (ES). ESS uses a combination of ideas based on existing search strategies and employs a novel $(1 + \sigma\lambda)$ ES implementation. It is essentially a succession of random searches, but by controlling the placement and distribution of these searches in a simple way, it proves possible to achieve comparable motion vector accuracy to the more established ME strategies, but with enhanced convergence speed.

Introduction: The goal of video codec design is to achieve the best perceived video quality, while at the same time minimising the amount of resource which is necessary to deliver it. The heart of most modern video coding techniques is the motion estimation (ME) algorithm, which minimises the temporal redundancy between images. Up to 60–80% of the computations in most established video codecs is taken up by the ME, motion compensation (MC) and transform coding, i.e. DCT or wavelet, and therefore even small enhancements in the efficiency of the ME can have significant effects on the overall performance of the codec. Full search (FS) provides excellent motion vector (MV) matches, but is far too computationally demanding. Therefore sub-optimal search strategies such as new three step search (NTSS) [1] and four step search (4SS) [2] were developed to provide a balance between quality and speed. The proposed algorithm, evolutionary strategy search (ESS), is developed with a view to enhance the ME in the British Broadcast Corporation's Dirac video codec [3]. ESS approaches the problem in a novel way, combining evolutionary computing [4] methods with aspects of the existing ME strategies, and is specifically designed to be simple to implement in hardware.

Proposed algorithm: Evolutionary strategy (ES) was first introduced in 1964 by Rechenburg and further developed by Schwefel at the University of Berlin [4]. The basic elements in ES are the definition of individuals, the control of parameters, fitness function, the terminating conditions and selection of best individual (BI).

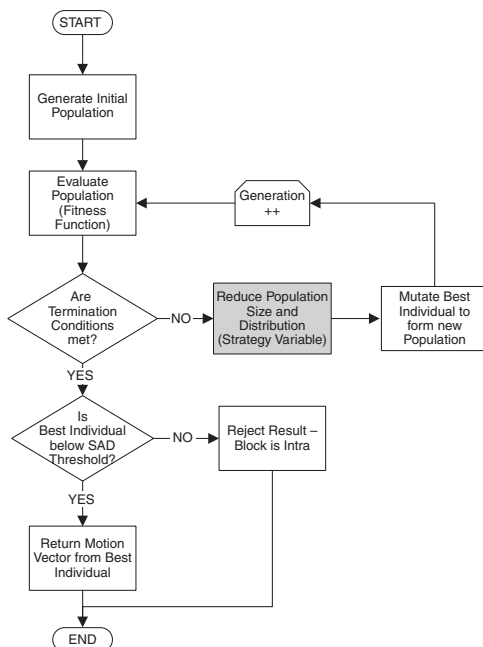


Fig. 1 Flow chart of ESS algorithm

The overall structure of the proposed ESS algorithm is illustrated in Fig. 1. In ESS, the initial population of individuals is uniformly distributed across the search area, which optimises the probability of there being an individual in the vicinity of the optimal solution. Each individual consists of three parts; two object variables, which are the candidate motion vectors (MV), X and Y , and the strategy variable, σ .

All individuals are evaluated by a fitness function, and the BI survives, while the remainder are cast aside. Since ESS is essentially $(1 + \lambda)$ ES implementation, the BI will be mutated and survive into the next generation cycle as a parent to form λ offspring (a.k.a. individuals). The process continues until the necessary termination conditions are met. The fitness function, in this case, is the sum of absolute difference (SAD) as:

$$\text{SAD}(X, Y) = \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} |A_{(i+X, j+Y)} - B_{(i,j)}|$$

where A and B are the macroblocks (MBs) under test and $[X, Y]$ is the candidate MV of a best match MB.

The innovation here is in the use of σ . In traditional ES, σ simply controls the distribution of the individuals. However in ESS, it is also used to control the population size; this novel ES implementation is represented as $(1 + \sigma\lambda)$. At each generation cycle the BI will tend towards the optimal solution, and therefore the size of the search area can be proportionally reduced if there is an improvement in the fitness value of the BI. In ESS the population size is also reduced by the same proportion, which means that at each generation the fitness function is performed proportionally less times, while the probability of achieving an optimal solution remains constant. Since the SAD requires S^2 operations per individual, a reduction in the population size has a significant effect on the number of operations overall and therefore the speed of the algorithm. Besides that, termination conditions, i.e. SAD threshold (TH) and generation number (GN), have been carefully designed to obtain the balance between the quality and speed of ESS. TH decides whether the BI has achieved a reasonable quality level. If not, it is rejected in favour of an intra-block. This setting essentially controls the acceptable quality of the results. The GN controls the number of iterations before the sequence is abandoned in favour of an intra-block. A larger GN will increase the probability of a good MV match, at the expense of increased time per MB. By varying these parameters it is possible to tailor the ESS to a particular application. For example, for video conferencing applications reducing the number of generations and increasing the SAD threshold will allow faster ME at the expense of some quality. However, it is important to note that the two are not independent, and must be adjusted in tandem.

Results: In the simulation, ESS is mainly compared with NTSS and 4SS since they are superior to other sub-optimal ME algorithms. All the ME algorithms are optimised in Matlab, and the testing is carried out on the same machine. The block size is fixed at 16×16 . As for the parameters of ESS, the initial population size, $\lambda = \text{sqrt}(\text{MB size})/2$, the TH = 128, and the GN = 4 throughout the testing. Results are presented using natural test sequences 'Motion1' and 'Kart'. The latter represents a significant challenge for the ME process owing to the frequent scene changes, multiple artefacts appearing/disappearing, and very high degree of irregular motion, pan and zoom. The results (Table 1, Figs. 2 and 3) demonstrate the performance advantage of ESS over FS, NTSS, and 4SS in speed and a comparable quality. Note that quality comparisons are made relatively to the FS, and speed comparisons are made relatively to the ESS since it is the fastest algorithm of all.

Table 1: Overall performance of ESS in comparison to FS, NTSS, 4SS

	Relative mean quality (bit/pixel)				Relative mean speed (sec/frame)		
	FS	ESS	NTSS	4SS	ESS	NTSS	4SS
Motion1	6.36	7.37	8.00	8.77	1.65	2.12	2.17
Kart	4.55	4.74	4.91	4.95	4.68	6.52	5.72

Motion1 is a standard definition (720×480 pixels) sequence specially designed by Streamcrest [5] to test the efficiency of the ME algorithms. It contains various artefacts which pose particular problems for ME algorithms. For the Motion1 sequence, ESS performs considerably faster than 4SS, with a better quality performance approximately 1.4 bit/pixel as shown in Fig. 2. ESS shows only slight variation in timing over the entire sequence compared to the other strategies, because it is constrained by the limit of four generations.

Kart is a high definition (1280×720 pixels) sequence supplied by the BBC R&D. It contains a Kart racing with a high degree of pan and

zoom, and several cut scenes. For the Kart sequence, ESS demonstrates comparable quality with greatly superior speed of convergence as shown in Fig. 3. It also does not suffer from the considerable timing variations of the other strategies, which are associated with the pan and cut scene elements of the sequence.

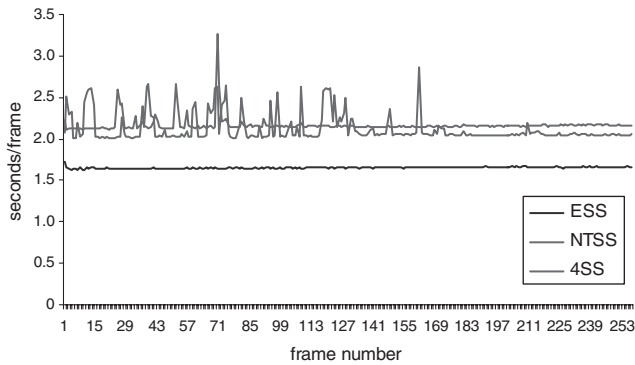


Fig. 2 Relative speed comparison of ESS in relation to NTSS and 4SS for 'Motion1' sequence

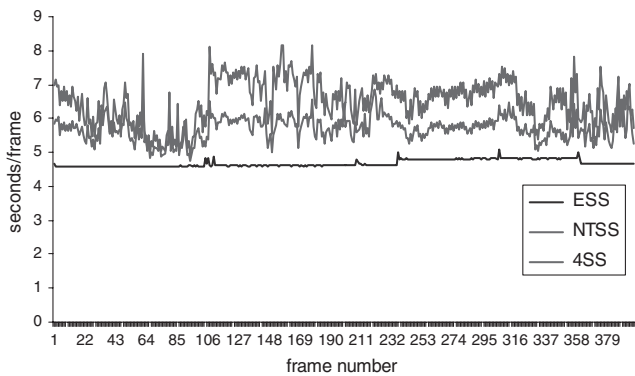


Fig. 3 Relative speed comparison of ESS in relation to NTSS and 4SS for 'Kart' sequence

Conclusions: A novel evolutionary strategy search algorithm is proposed. The results indicate comparable quality to the established ME algorithms with enhanced convergence speed especially when applied to larger, high pan, high zoom video sequences. The simplicity of the design makes it suitable for hardware implementation and provides a predictable timing profile, which is essential for encoding live video.

© The Institution of Engineering and Technology 2006
 5 April 2006
 Electronics Letters online no: 20061079
 doi: 10.1049/el:20061079

J. Hay and K.K. Loo (School of Engineering and Design, Brunel University, Uxbridge UB8 3PH, United Kingdom)
 E-mail: Jonathan.Loo@brunel.ac.uk

References

- 1 Li, R., Zeng, B., and Liou, L.: 'A new three-step search algorithm for block estimation', *IEEE Trans. Circuits Syst. Video Technol.*, 1994, **4**, pp. 438–442
- 2 Po, L.M., and Ma, W.C.: 'A novel four-step search algorithm for fast block estimation', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6**, pp. 313–317
- 3 BBC's Dirac Video Codec [online]. Available at URL: <http://dirac.sourceforge.net/> (accessed March 2006)
- 4 Schwefel, H.-P.: 'Evolution and optimum seeking' (John Wiley & Sons, New York, 1995)
- 5 Streamcrest Motion1 Test Sequence [online]. Available at URL: <http://www.streamcrest.com/mp4.shtml> (accessed March 2006)