

# AN EVOLUTIONARY STRATEGY BASED MOTION ESTIMATION ALGORITHM FOR H.264 VIDEO CODECS

S. Adedoyin, W.A.C. Fernando, H. Kodikara Arachchi, T. Kalganova, K.K. Loo  
School of Engineering and Design, Brunel University, UB8 3PH, United Kingdom

Email {ee01sba, anil.fernando, hemantha.kodikara-arachchi, tatiana.kalganova, jonathan.loo}@brunel.ac.uk

## Abstract

*In this paper, we propose a new motion estimation algorithm based on evolutionary strategy (ES) for the H.264 video codec applied to monoscopic video. The proposed technique applies in macroblock basis and performs a parallel local search for the motion vector associated with the minimum motion compensated residue. For this purpose ( $\mu+\lambda$ )-ES is used with heuristically and randomly generated population of initial motion vectors. Experimental results show that the proposed scheme can reduce the computational complexity up to 50% of the motion estimation algorithm used in the H.264 reference codec at the same picture quality. Therefore, the proposed algorithm provides a significant improvement in motion estimation in the H.264 video codec.*

## 1. Introduction

The new H.264 video coding standard [1] developed jointly by the Video Coding Experts Group (VCEG) of ITU and the Motion Picture Experts Group (MPEG), i.e. Joint Video Team (JVT), provides a versatile coding format, accommodating a wider variety of bandwidth requirements, picture formats and unfriendly network environments that may possess high jitter, packet loss, and bandwidth instability. Although the features of H.264 make it viable for many applications, many alterations, extensions and optimisations have to be made before it can be applied to such systems. The vastly improved coding efficiency of H.264 comes at a price of complexity, which may not be realizable for real time transmission. In order to be attractive enough for it to be adopted by commercial organisations/industry, it is required to encode in real-time. Unfortunately due to the stringent requirements governing the design and operational simplicity of standardized H.264 decoder and the optimisations thus involved, increases the encoder complexity beyond real-time capability. Motion estimation is the most complex part and it takes about 40-60% computational cost of the encoder.

In this paper, we propose a new motion estimation algorithm based on evolutionary strategy (ES) for the H.264 video codec. Evolutionary strategy applied to motion estimation is a kind of evolutionary algorithm where individuals (potential solutions) are encoded using  $x$  and  $y$  coordinates. Each coordinate is described by two different types of genes: object and strategy genes. Strategy gene determines the degree of mutation to be applied to the corresponding object gene in the coordinate. The value of object gene to be mutated depends on the strategy gene value.

The rest of the paper is structured as follows. Section 2 describes the motion estimation process. Section 3 introduces proposed ES-based motion estimation algorithm for the H.264 codec. Section 4 presents simulation results and section 5 concludes the paper.

## 2. Motion Estimation (ME)

One of the key elements of video compression schemes is motion estimation. For a video, sequence consists of a series of frames, to achieve compression, the temporal redundancy between adjacent frames can be exploited. That is, a frame is selected as a *reference*, and subsequent frames are *predicted* from the reference using a technique known as motion estimation. The process of video compression using motion estimation is also known as interframe coding. When using motion estimation, an assumption is made that the objects in the scene have only translational motion. This assumption holds as long as there is no camera pan, zoom, changes in luminance, or rotational motion. However, for scene changes, interframe coding does not work well, because the temporal correlation between frames from different scenes is low. In these cases, a second compression technique is used, known as intraframe coding.

In a sequence of frames, the current frame is predicted from a previous frame known as reference frame. The current frame is divided into macroblocks, typically  $16 \times 16$  pixels in size. This choice of size is a good trade-off between accuracy and computational cost. However, different motion estimation techniques may choose different block sizes, and may vary the size of the blocks within a given frame. Each macroblock is compared to a macroblock in the reference frame using some error measure, and the best matching macroblock is selected. The search is conducted over a predetermined search area. A vector denoting the displacement of the macroblock in the reference frame with respect to the macroblock in the current frame is determined. This vector is known as a motion vector. When a previous frame is used as a reference, the prediction is referred to as forward prediction. If the reference frame is a future frame, then the prediction is referred to as backwards prediction. Backwards prediction is typically used with forward prediction, and this is referred to as bidirectional prediction.

In video compression schemes that rely on interframe coding, motion estimation is typically one of the most computationally intensive tasks, taking up to 60% of the processing time. During reconstruction, the reference frame is used to predict the current frame using the motion vectors. This technique is known as motion compensation. During motion

compensation, the macroblock in the reference frame that is referenced to by the motion vector is copied into the reconstructed frame. In motion estimation, the search process can be modified to suit the needs of a particular algorithm. The search area is typically restricted to lower the computational cost associated with block matching. Also, in many cases, the objects in the scene do not have large translational movements between a frame and the next. That is, the fact that frames in a video sequence are taken at small intervals of time is exploited. Many techniques have been proposed to solve the problem of determining the best match at the lowest computational cost [2-5].

### 3. Proposed H.264 Encoder With ES-Based ME

Evolutionary computation (EC) theories were developed originally from observing natural evolution of life form. Because of this, the terminology surrounding the field of EC is full of analogies with natural evolutionary process. It was particularly from Darwin's theories [6] that the best techniques regarding the optimization, modelling and the control of unknown processes were developed. EC has long been exploited in the video coding field. A very well known form of EC called Genetic Algorithm (GA) was used to perform image registration as part of a larger Digital Subtraction Angiography (DAS) system [7, 8]. Subsequently, GA search algorithm has been applied for motion estimation [9-12]. Hardware implementation of Four-Step genetic search algorithm was proposed in [9].

Similarly, the Evolutionary Strategies (ES) were developed to solve technical optimization problems in video coding field. Thus, the motion and disparity estimation has been carried out using a  $(1+\lambda)$  rudimentary ES for stereoscopic video sequences, which includes calculation of P- and B-frames, weighted prediction, joint motion disparity estimation [13,14]. In this paper, we apply ES to estimate the motion vectors in H.264 based video codec for monoscopic video sequences.

### 3.1 Proposed evolutionary strategy

ES typically uses deterministic selection in which the worst solutions are purged from the population based directly on their fitness function value. The  $(\mu+\lambda)$ -Evolutionary Strategy demonstrated in Figure 1 is used in this work with an increasing level of imitation of biological evolution [15], where  $\mu$  means the total number of parents in previous population, and  $\lambda$  stands for the number of offspring generated from mutated parents.

### 3.2 Chromosome representation

Each chromosome represents the data for coordinates  $x$  and  $y$ . Each coordinate is described by 2 genes: object and strategy genes as shown in the Figure 2 (a). Object gene defines the actual coordinate in the image. The value of object gene is determined from the search window size. For example, if the search window is within the range  $[-16, 16]$ , then the value of object gene can take any integer number inside of this range. The search window size depends on the maximum motion vector size. Strategy gene determines whatever local or global search will be carried out. Smaller value of strategy gene more localised become the search process. The negative value defines the decrement of mutated gene and positive values respectively determine the increment of the mutated gene. The strategy parameter depends on the window size and can take any value up to its maximum. In order to implement the local search, we choose to set the strategy parameter to values  $-1$  or  $1$ .

### 3.3 Fitness Function

The quality of the chromosome is defined by fitness function. Fitness function is calculated based on the Sum of Absolute Difference (SAD). Each chromosome in newly generated population is evaluated using fitness function.

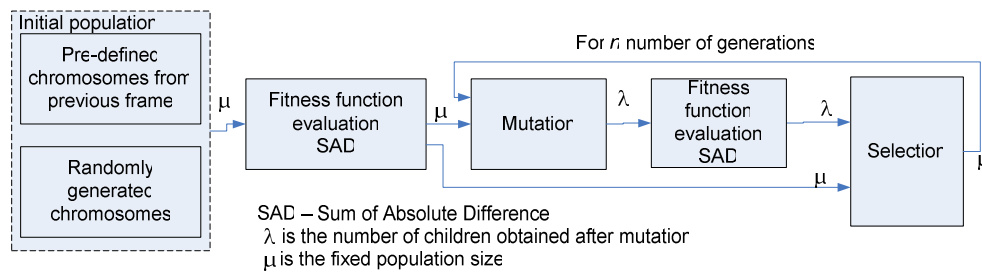


Figure 1  $(\mu+\lambda)$ -Evolutionary Strategy-based motion estimation algorithm

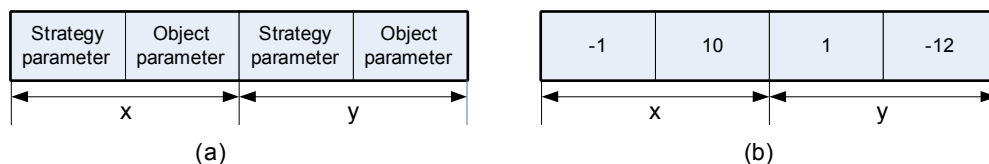


Figure 2 Chromosome representation (a) General representation of chromosome; (b) an example of chromosome

### 3.4 Evolutionary strategy operators

In order to reduce the number of generations required to obtain the satisfactory solution, the initial population is generated from both pre-defined and randomly generated chromosomes. The pre-defined chromosomes are determined based on the knowledge from the previously coded macro blocks located in the previous and the current frame. The remaining chromosomes have been generated randomly taking into account the search window size, specified in advance. In our case the search window size was set to 16x16. In the experimental results we have used 10 individuals that are defined from the previous frame and 10 chromosomes have been generated randomly. Therefore the initial population size used in the experimental work was set to 20.

Let us consider in detail the procedure on how the pre-defined chromosomes have been generated. Firstly restrictions were placed upon the chromosomes in order to generate points from the surrounding cover block location. In other words, for the block located at (0, 0) in the current frame, we would define 9 surrounding positions with increment equalled to 1. Thus, the following points will be defined: (-1, 1), (-1, 0), (0, 1), etc. The strategy gene for each coordinate is defined randomly. The 10<sup>th</sup> chromosome is the motion vector which is predicted from the already encoded surrounding macroblocks [1].

Selection takes place only amongst the offspring's (mutated values) and parents. The size of population in the next generation is fixed and set to 20 individuals. The new population is generated from the 20 best chromosomes from the previous population that combines both parents and offspring as shown in Figure 1.

Mutation rate defines the percentage of genes to be mutated in a newly generated population. Mutation rate used in the experiments was set to 8.5%. In general the value of the strategy gene is generated randomly from the local search increment window specified in advance. In our case, the strategy gene value can vary within the following range [-1, 1]. For example, if the strategy gene in the  $x$  coordinate shown in Figure 2 (b) is chosen to be mutated, then its value will be changed to 1. The new value of the object gene (if this gene has been chosen to be mutated) is defined as following:

$$x_{op}^{new} = x_{op} + x_{sp} \quad [1]$$

where  $x_{op}^{new}$  is the new value of mutated object gene for  $x$  coordinate,  $x_{op}$  and  $x_{sp}$  are the object and strategy genes for  $x$  coordinate respectively. Similarly the genes for  $y$  coordinate are calculated.

Let us consider an example shown in Figure 2 (b). The value of  $x$  coordinate is 10, if this gene has been chosen to participate in mutation, the value of this gene will be decremented by 1 and a new gene will be produced with an object gene equalled to 9. In the case of the  $y$  coordinate with an object parameter of

value -12, this will be incremented by one, if it is chosen to be mutated and therefore will produce a new gene with an object gene value of -11. If object genes for both coordinates have been chosen to be mutated then the following offspring with unchanged strategy genes will be produced: (-1, 9)(1,-11).

Mutation rate defines the percentage of genes to be mutated in the population. In the current work, 8.5% of the genes randomly selected from the population are mutated.

## 4. Results and Discussion

The proposed motion estimation algorithm was implemented into H.264 reference encoder version JM 9.3 and tested with various sequences using a high spec PC. We used the existing motion estimation algorithm (spiral search) in JM 9.3 reference software to compare the performance with the proposed scheme. Figure 3-Figure 6 illustrate the PSNR comparison with bpp (bits/pixel) for *Salesman*, *Suzie*, *Mother and Daughter* and *Carphone* sequences respectively. Results show that both algorithms produce a similar picture quality for all the sequences. Table 1 presents the processing time for both algorithms with the sequences considered. It is clear that the proposed scheme saves the processing time up to 50% compared to the spiral search algorithm. We tested the proposed algorithm with other video sequences and observed a similar behaviour.

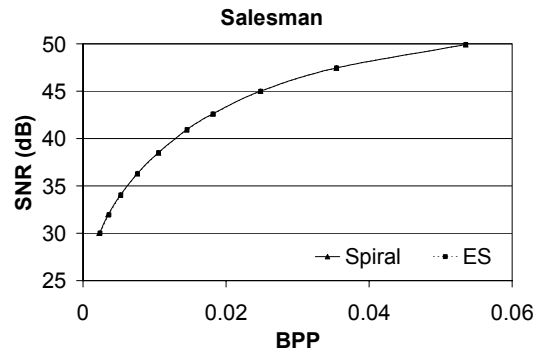


Figure 3. Objective quality comparison for the *Salesman* sequence

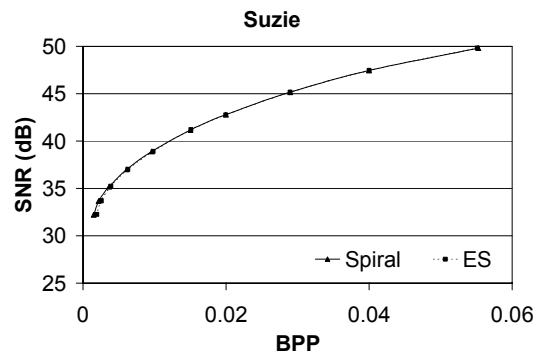


Figure 4. Objective quality comparison for the *Suzie* sequence

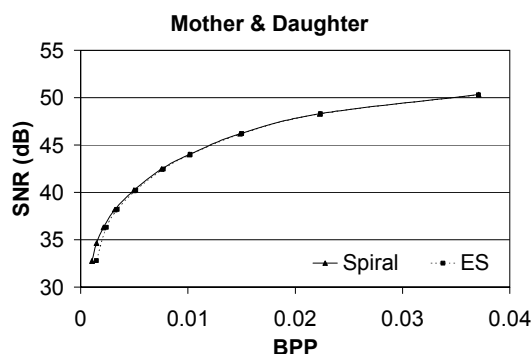


Figure 5. Objective quality comparison for the *Mother and Daughter* sequence

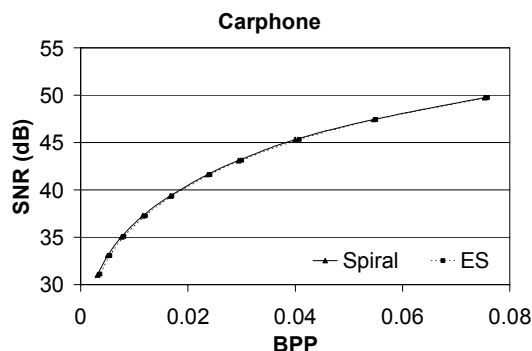


Figure 6. Objective quality comparison for the *Carphone* sequence

Table I. Coding complexity comparison between the spiral search and the proposed ES search

Sequence	Average coding time per frame (s)		Coding complexity improvement (%)
	Spiral Search	ES Search	
<i>Mother and Daughter</i>	0.3201	0.1693	47.11
<i>Foreman</i>	0.3073	0.1732	43.64
<i>Carphone</i>	0.3345	0.1672	50.01
<i>Container</i>	0.3192	0.1593	50.09
<i>Suzie</i>	0.3136	0.1642	47.64
<i>Salesman</i>	0.3234	0.1681	48.02

## 5. Conclusions

In this paper, we proposed an  $(\mu+\lambda)$ -evolutionary strategy based ME algorithm for the H.264 encoder. Experimental results show that the proposed scheme can reduce the computational complexity up to 50% compared to the spiral search algorithm with the same picture quality. Therefore, the proposed algorithm provides a significant improvement in the H.264 video encoder. We are planning to extend the proposed scheme to process multiple reference frames.

## References

[1] T. Wiegand and G. Sullivan, "Final draft international standard (FDIS) of joint video specification (ITU-T rec.

H.264 ISO/IEC 14 496-10 AVC)," in Joint Video Team (JVT) of ITU-T SG16/Q15 (VCEG) and ISO/IEC JTC1/SC29/WG1, Annex C, Pattaya, Thailand, Doc. JVTG050, Mar. 2003.

- [2] Lai-Man Po, Wing-Chung Ma, "A Novel Four Step Search Algorithm For Fast Block Motion Estimation" IEEE Transactions on Circuits and Systems for VideoTechnology, vol. 6, no.3, pp 313-317, June, 1996.
- [3] <http://atlantis.ucc.ie/dvideo/contents.html>
- [4] Zahariadis, T.; Kalivas, D., "Fast algorithms for the estimation of block motion vectors", Electronics, Circuits, and Systems, ICECS '96., Proceedings of the Third IEEE International Conference on ,Volume: 2 ,13-16 Pages:716 - 719 vol.2 2. October, 1996.
- [5] M. Ghanbari, "The cross search algorithm for Motion Estimation" IEEE Transactions on Communications, vol. 38, no. 7 pp 950-3, July, 1990.
- [6] Charles Darwin, The Origin of Species: By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life (Bantam Classic), Bantam Classics, Reprint. 1999.
- [7] Fitzpatrick, J.M., Grefenstette, J.J. and Van-Gucht, D. "Image registration by genetic search," Proceedings of Southeastcon 84, Louisville, KY, 460-464, Apr 1984.
- [8] Grefenstette, J.J. and Fitzpatrick, J.M. "Genetic search with approximate function evaluations," Proc. Intl. Conf. on Genetic Algorithms and their Applications, Pittsburgh, PA, 112-120, Jul. 1985.
- [9] Man F. So & Angus Wu, "Hardware Implementation of Four-Step Genetic Search Algorithm", IEEE Signal Processing Society 1999 Workshop on Multimedia Signal Processing, Copenhagen, Denmark, September 1999.
- [10] Xu Yuelei, Bi Duyan and Mao Baixin, "A Genetic Search Algorithm For Motion Estimation", *Proceedings of 5th International Conference on Signal Processing Proceedings*, Beijing, China, 2000.
- [11] Guanghua Qiu, Chaohuan Hou, "A New Fast Algorithm for the Estimation of Block Motion Vectors", *Proceedings of 3rd International Conference on Signal Processing*, Beijing, China, 1996.
- [12] Shen Li, Wei-pu Xu, Hui Wang, Nan-ning Zheng, "A Novel Fast Motion Estimation Method Based on Genetic Algorithm", *Proceedings of International Conference on Signal Processing, Kobe, Japan, 1999.*
- [13] K. Ponudurai, W.A.C. Fernando and K.K. Loo, "Joint Motion and Disparity Estimation in Stereo Video Sequences Using Evolutionary Strategy," *Proceedings of 1st Regional Conference on ICT and E-Paradigms*, Colombo, Sri Lanka, 2004.
- [14] K. Ponudurai, W.A.C. Fernando and K.K. Loo, "Joint Motion and Disparity Estimation in Stereo Video Sequences Using Evolutionary Strategy," *Proceedings of The Seventh International Symposium on Wireless Personal Multimedia Communications*, Abano Terme, Italy, September 2004.
- [15] Ingo Rechenberg, *Evolutionsstrategie '94*. Stuttgart: Frommann-Holzboog 1994.