Signal Processing: *Image Communication* 39 (2015) 121–140

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A novel Cooperative Motion Estimation Algorithm based on Particle Swarm Optimization and its multicore implementation

Manal K. Jalloul *, Mohamad Adnan Al-Alaoui

American University of Beirut, Faculty of Engineering and Architecture, Electrical and Computer Engineering Department, P.O.Box 11-0236/ECE Riad El-Solh, Beirut 1107 2020, Lebanon

ARTICLE INFO

Article history:
Received 3 November 2014
Received in revised form
11 September 2015
Accepted 11 September 2015
Available online 25 September 2015

Keywords:
Motion estimation
Cooperative multi-swarm
PSO
Parallel processing

ABSTRACT

A cooperative motion estimation (ME) scheme using a modified Particle Swarm Optimization (PSO) algorithm is presented. The proposed algorithm is based on a multi-swarm PSO model where a swarm of PSO particles is defined for each macroblock (MB) in the frame. Motion estimation is then performed in a cooperative manner concurrently for all the MBs in the frame. Cooperation between neighboring MBs during the motion estimation process is allowed through a communication step to exchange information about the motion vectors found so far in the estimation process. This synergic relationship between the swarms of adjacent MBs allows refining the motion search and leads to both a faster convergence of the PSO process and an improvement in the resulting motion vectors. Several techniques are also proposed to improve the search capacity and computational complexity of the PSO iterations. A novel PSO initialization scheme that exploits the existing temporal correlation is proposed to remove dependency between adjacent MBs. A fitness function history preservation mechanism is also presented to prevent redundant repeated calculations of the fitness function of a given search point by the PSO particles which dramatically decreases the computational complexity. Moreover, the maximum allowed velocity of the particles is adaptively varied during the PSO iterative process which provides a balance between search exploration and exploitation. The proposed scheme exhibits a high level of data parallelism since it is capable of performing motion estimation for all the MBs of the frame in parallel rather than serially. As a result, the presented algorithm is amenable to parallel processing techniques. In this paper, a multicore implementation of our proposed algorithm is performed using the MATLAB® Parallel Computing Toolbox™ (PCT). Extensive simulations are performed to analyze the performance of the presented algorithm. It is found that the presented scheme provides improvements in terms of accuracy and computational complexity as compared to conventional fast motion estimation techniques and two state-of-the-art PSO-based ME schemes. An analysis of the parallel performance shows that the presented scheme is highly scalable and that the parallel efficiency increases with the increase in video resolution. The multicore implementation of the proposed algorithm using MATLAB could achieve a speedup of 6.21 on eight CPU cores for high-definition (HD) video sequences. The multicore performance of the proposed scheme is also compared with existing parallel algorithms in the literature and is shown to give superior results.

* Corresponding author. Mobile: 961 3089040.
E-mail addresses: Mkj00@aub.edu.lb (M.K. Jalloul), adnan@aub.edu.lb (M.A. Al-Alaoui).

http://dx.doi.org/10.1016/j.image.2015.09.010
0923-5965/

1. Introduction

The high correlation between successive frames of a video sequence makes it possible to achieve high coding efficiency in a video coding system by reducing the temporal redundancy. Motion compensated video coding technique, which predicts current frame from previous frame (or reference frame), has been used to exploit the temporal redundancy between successive frames. Motion estimation (ME) plays an important role in such an inter-frame predictive coding system. Among many types of motion estimation algorithms, block matching technique has been adopted in many video compression standards, such as MPEG 1 [1], MPEG 2 [2], H.261 [3], H.263 [4], H.264 [5], and the latest H.265/HEVC [6] due to its simplicity.

In the block-matching technique, frames are divided into blocks and one motion vector is associated with each block. For each block in the current frame, the motion estimation searches for a motion vector which points to the best matching block in the reference frame. The best matching block is then used as the predictor for the current block. Although the exhaustive search (ES) block-matching algorithm is the simplest, it is computationally very intensive. It provides an optimal solution by exhaustively evaluating all the possible candidates within the search range in the reference frame. There is a growing need to decrease the computational burden of the ES scheme while preserving the quality of the produced motion information. The motion estimation process needs to be accelerated to meet the real time processing requirements of several cutting-edge multimedia applications.

In the literature, many fast motion estimation searching methods were researched to reduce the computational cost of the ES method. These algorithms included Three-Step Search (3SS) [7], New Three Step Search (NTSS), Simple and Efficient Three Step Search (SETSS) [8], Four-Step Search (4SS) [9] which can be generalized to *N*-step search (NSS), Diamond Search (DS) [10], Cross-Diamond Search (CDS) method [11], Adaptive Rood Pattern Search (ARPS) [12], and the Hexagon-based search [13]. In each of these fast searching methods, a different search pattern is employed to reduce the number of search points. These algorithms reduce the computational complexity with negligible loss of image quality only when the motions matched the pattern well; otherwise, the image quality will decrease. In [14], a hybrid Unsymmetrical Multi-Hexagon-grid search (UMHexagonS) algorithm, which attempt to use many search patterns, has achieved both fast speed and good rate-distortion performance. Although uneven search patterns are used to meet the assumption that motion is more horizontal than vertical, it cannot adaptively choose the intensive search area for irregular motions. To tackle this drawback, Predictive Intensive Direction Searching (PIDS) algorithm [15] was developed. In PIDS, the correlation of predicted MV and optimal MV are studied. On the basis of MV prediction information, the area with high correlation is intensively searched, while other areas are coarsely searched. PIDS successfully speeds up the process compared to UMHexagonS. However, this algorithm still searches each direction exhaustively, which may cause searching resource waste. In [16], a novel Predictive Priority Region Search (PPRS) algorithm that performs adaptively search in direction and locality regions was proposed. Other FME algorithms proposed in the literature include motion adaptive search (MAS) [17] which utilize the motion activity information to adjust the search strategy, Variable Step Search (VSS) algorithm [18] which employs correlation between neighboring motion vectors to determine motion search range, and the Multi-Path Search (MPS) algorithm [19] in which all the eight neighbors around the origin of the search window are used to find candidate points. In addition to the above, several high efficiency algorithms were presented in the literature for ME. These algorithms significantly reduce the number of checking points examined while retaining the video quality. These techniques accomplish this by initially considering several highly likely predictors, introducing very reliable early-stopping criteria to terminate the search at any checking point, and using very efficient checking patterns for optimizing and improving the search even further. These algorithms include the Motion Vector Field Adaptive Search Technique (MVFAST) [20], the Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) [21], the Advanced Predictive Diamond Zonal Search (APDZS) [22], and the Enhanced Predictive Zonal Search (EPZS) [23].

Block matching motion estimation can be formulated into an optimization problem where one searches for the optimal matching block within a search region which minimizes a certain block distortion measure (BDM), which is usually taken as the sum of absolute difference (SAD). Such a problem is classified as non-convex since the objective function is multimodal and has many local minima. The above fast block matching methods suffer from poor accuracy since they dictate that only a very small fraction of the entire set of candidate blocks be examined, thereby making the search susceptible to being trapped into local optima on the error surface. The underlying theory of these search engines comes from the idea that the block distortion measure reduces monotonously when search points move from the farthest point toward the optimal point. In practice, applications do not always completely obey the monotonous rule. Therefore, these fast search engines are easily trapped into the local optimal solutions and miss the global optimal solution. In order to escape from the problem of local minima, several approaches were recently presented in the literature to use modern global optimization algorithms to solve the problem of motion estimation. In [24,25], the Genetic Algorithm (GA) has been considered for motion estimation. The proposed algorithms, however, tend to be complex and suffer from a high computational burden. In [26], the Simulated Annealing (SA) concept is employed to control the searching process and to adaptively choose the intensive search region. In addition, artificial bee colony optimization (ABC) [27] and differential evolution (DE) [28] were also proposed for motion estimation.

Recently, there have been some attempts in the literature to apply Particle Swarm Optimization (PSO) to solve the problem of ME [29–36]. The PSO-based motion estimation methods introduced in [29–33] either have higher computational complexity [29] or have lower estimation