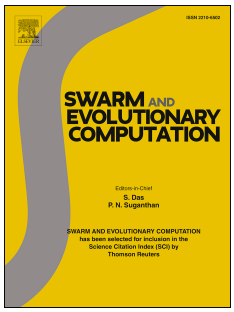
Accepted Manuscript



Agent-based game theoretic model for block motion estimation and its multicore implementation

Manal K. Jalloul, Mohamad Adnan Al-Alaoui

PII: S2210-6502(17)30144-X

DOI: 10.1016/j.swevo.2018.02.012

Reference: SWEVO 365

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 1 March 2017

Revised Date: 28 September 2017

Accepted Date: 23 February 2018

Please cite this article as: M.K. Jalloul, M.A. Al-Alaoui, Agent-based game theoretic model for block motion estimation and its multicore implementation, *Swarm and Evolutionary Computation BASE DATA* (2018), doi: 10.1016/j.swevo.2018.02.012.

Agent-based game theoretic model for block motion estimation and its multicore implementation

Manal K. Jalloul and Mohamad Adnan Al-Alaoui

Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

Abstract

Motion estimation (ME) is one of the main tools employed for eliminating temporal redundancies in video coding. It is the most critical and time-consuming tool of the complete encoder and typically requires 60%–80% of the total computational time. Block-matching ME (BME) algorithms divide a frame into macroblocks (MB) and look for the best possible match in the reference frame. This paper introduces a novel parallel framework to speed up the BME process. This is done by introducing a novel level of parallelism within the MB. The problem of BME is cast in a non-cooperative game-theoretic setting and a distributed multi-agent system is employed to solve the problem. First, a given MB is divided into subblocks and an agent is defined for each subblock. Then, the problem is formulated as a Consensus game and our approximation of the global utility function for the MB is defined. Building on this, agents' utilities are derived so that the resulting game is a potential game. To solve the game, distributed sequential and simultaneous algorithms based on game-theoretic Best Response Dynamics (BRD) and particle swarm optimization (PSO) are presented. Each agent uses PSO as its local search engine to autonomously maximize the utility of its subblock and BRD drive the agents with minimum local communication towards the maximum of the global utility function of the whole MB. Experimental results show that these algorithms provide good estimation quality with low computational cost as compared to other techniques. Moreover, in addition to its decentralized and distributed nature, the simultaneous algorithm is also inherently parallel at the agents' level within the MB. A parallel implementation of this algorithm using the MATLAB Parallel Computing Toolbox™ (PCT) on a multicore system shows that speedup is indeed obtained.

Keywords

motion estimation; game theory; potential games; parallel processing; video coding; multicore based parallel framework; particle swarm optimization; multi-agent system; parallel implementation; MATLAB

1 Introduction

In the entire motion based video compression process, motion estimation is the most computationally expensive and time-consuming process. Motion estimation involves interframe predictive coding, one of the most powerful image coding techniques, which calculates motion vectors and can eliminate redundancy in natural scenes. Research in the past few decades has focused on reducing both of these side effects of motion estimation.

Block matching techniques are widely used motion estimation methods to obtain the motion compensated prediction. By splitting each frame into macroblocks, motion vector of each macroblock is obtained by using block matching algorithm (or motion estimation algorithm). In order to get motion vector of each macroblock, the most obvious and simplistic method is the exhaustive search (ES) algorithm. All possible displacements in the search window are evaluated using block-matching criteria (cost function). The advantage of the ES algorithm is that we can find the absolute optimal solution. However, its high computational complexity makes it impossible for real-time implementation. Because the computational complexity of video compression, the compression efficiency and the compression quality are determined by the motion estimation algorithm, development of fast motion estimation algorithms for real-time applications becomes compelling.