Information Sciences 296 (2015) 322–344

Contents lists available at ScienceDirect

Information Sciences

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

Massive surveillance data processing with supercomputing cluster

A. Czyżewski, P. Bratoszewski, A. Ciarkowski *, J. Cichowski, K. Lisowski, M. Szczodrak, G. Szwoch, H. Krawczyk

Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, ul. Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Article history:
Received 22 March 2013
Received in revised form 3 November 2014
Accepted 7 November 2014
Available online 15 November 2014

Keywords:
Smart surveillance
Massive video data processing
Object detection
Parallel processing

ABSTRACT

In recent years, increasingly complex algorithms for automated analysis of surveillance data are being developed. The rapid growth in the number of monitoring installations and higher expectations of the quality parameters of the captured data result in an enormous computational cost of analyzing the massive volume of data. In this paper a new model of online processing of surveillance data streams is proposed, which assumes the use of services running within a supercomputer platform. The study presents some of the highly parallelized algorithms for detecting safety-threatening events in high-resolution-video streams, which were developed during the research, and discusses their performance on the supercomputer platform.

1. Introduction

Modern video surveillance systems are composed of a large number of cameras covering wide areas. It is difficult for the system operator to notice every important event. Many existing installations are based only on video recording for the post-factum offline analysis. Automatic event detection improves the performance of the monitoring system by providing notifications on detected events that may potentially be security threats and that require the operators attention [15]. Automatic analysis of video streams and event detection are subjects of many scientific studies [16]. However, analysis of even a single visual data stream with complex algorithms is computationally intensive. In cases where multiple data streams, carrying various modalities, are analyzed, the computational requirements increase substantially. Standard computers have limited resources and they are only able to analyze data from a small number of video sources, with a limited number of computationally intensive algorithms. In practical applications, video streams from a large number of cameras have to be analyzed at the same time. Moreover, analysis of each video stream cannot be limited to one algorithm, because different types of complex analysis (e.g. object detection, people counting, intrusion detection, etc.) have to be performed at the same time in online mode. Achieving this goal on a typical computer is not possible. Additionally, if separate machines analyze individual video streams, there is a problem of integration of partial analysis results in order to perform a complex event detection in the whole observed area. In order to cope with these requirements it is necessary to employ an efficient platform equipped with adequate resources and capable of parallelizing the key computations.

* Corresponding author.
E-mail addresses: andcz@sound.eti.pg.gda.pl (A. Czyżewski), bratoszewski@sound.eti.pg.gda.pl (P. Bratoszewski), rabban@sound.eti.pg.gda.pl (A. Ciarkowski), jay@sound.eti.pg.gda.pl (J. Cichowski), lisowski@sound.eti.pg.gda.pl (K. Lisowski), szczodry@sound.eti.pg.gda.pl (M. Szczodrak), greg@sound.eti.pg.gda.pl (G. Szwoch), hkrawk@eti.pg.gda.pl (H. Krawczyk).

http://dx.doi.org/10.1016/j.ins.2014.11.013
0020-0255/

In this article we address the problem of automated, near-realtime, analysis of massive surveillance data from multiple video cameras and using various complex analysis algorithms. We propose utilizing the computing power of a supercomputer platform for this task. Contrary to standard computer machines, a supercomputer platform provides resources (both the processing power and memory storage) required for concurrent and efficient processing of data from a large number of sensors in the surveillance system, employing a set of computationally complex algorithms. Single supercomputer nodes may be dedicated for processing single source data with a resource-demanding algorithm, while other nodes may integrate partial results from multiple sources running less complex algorithms. Such a system requires important problems to be solved relating to algorithm implementation, data exchange, resource management, etc., but it also provides an efficient solution for the analysis of massive amounts of surveillance data. To date, there have been no published works related to implementing a system for complex video analysis and event detection on a supercomputer platform.

Implementation of the system described above on a supercomputer platform requires two important problems to be solved: implementation of individual algorithms on supercomputer nodes and efficient data exchange between the algorithms running on separate nodes. The latter issue was addressed within the project "MAYDAY EURO2012 – Supercomputer Platform for Context Analysis of Data Streams in Identification of Specified Objects or Hazardous Events", which proposes a new model of surveillance data processing, based on stream analysis services running within a dedicated hardware and software environment. Contrary to typical contemporary installations, the MAYDAY project not only deals with visual surveillance but also supports other modalities, like audio or thermovision, including fusion of these signals – multimodal analysis. Nevertheless, in this paper we focus primarily on the subject of video data processing and we refer to the literature regarding other aspects of the MAYDAY project (e.g. [28,35,36]). A low-level software layer for algorithm implementation and data exchange was realized within this project in the form of the KASKADA framework, which utilizes the message transmission interface for data exchange between the nodes and provides an application development interface for the algorithm developers. Though implementation of this framework on the supercomputer platform lies outside the scope of this paper, some details may be found in the literature [30–32].

There are two main goals for this paper. The first of them is to demonstrate how a supercomputer platform may be utilized for processing massive data from the surveillance systems in an efficient way. For this purpose, the framework for multi-sensor data processing on a supercomputer platform is provided with some typical, computationally demanding algorithms for video analysis, example scenarios of video analysis in practical applications are presented and algorithms required for realizing these scenarios are described. It should be stressed that practical applications of the proposed system are not limited to utilizing the KASKADA framework and the algorithms presented here. Any framework which allows for an efficient data exchange between the supercomputer nodes may be used as the low-level layer and also other algorithms than those presented here may be implemented within the system.

The second goal is to present how the most commonly used video analysis algorithms, such as background subtraction or optical flow, may be efficiently implemented on multi-core supercomputer nodes for parallel data processing. Efficient video processing requires a knowledge of the computational resources needed by each algorithm. Supercomputer resources are expensive, so it is essential to assign only the really needed resources to each algorithm and to provide the means to adjust them dynamically according to the video features. For example, several low-resolution video streams may be processed on a single node, while the high-definition video stream may need to be divided into two or more nodes in order to achieve a near-realtime performance. Therefore, each of the individual video processing algorithms described here is implemented in the parallel form on a single supercomputer node and their performance in processing video streams of varying resolution is tested. These experiments provide knowledge of the number of processing cores needed for near-realtime processing of a given video of a particular resolution by each individual algorithm.

The remainder of this paper is organized as follows. Section 2 briefly describes the general framework within which the video analysis algorithms are run. Section 3 is dedicated to details of each video processing algorithm and methods of their parallel implementation on the supercomputer node. The results of performance testing of each algorithm are presented and discussed in Section 4 and practical use cases of the proposed system are described in Section 5. The paper ends with conclusions and suggestions for future research.

2. Framework for massive video analysis

Modern video surveillance systems are composed of a large number of cameras, many of which are of high resolution. Most of the algorithms that perform analysis of video streams for the detection of important events are computationally complex. This is most evident for algorithms that process the source camera images pixel-by-pixel, for example, a background subtraction performed for detection of moving objects. Therefore, the computation time increases with image resolution. A camera of 1920×1080 pixels providing images at 15 frames per second (fps) requires the algorithm to perform more than 31 million pixel operations per second. Therefore, for a sequential processing, video analysis in the online mode using a typical personal computer is possible only for low to medium resolution streams and merging the analysis results from various sources and algorithms is usually problematic. In order to avoid such problems, a supercomputer platform is used to optimize the analysis by distributing the processing into multiple nodes that run the algorithms in parallel. Such a platform has to be able not only to process the data, but also to transfer the data among the algorithms and store them in the memory in an efficient way. In the case of 1920×1080 resolution cameras, an uncompressed (raw) image frame in