REAL-TIME BLEEDING DETECTION IN GASTROINTESTINAL TRACT ENDOSCOPIC EXAMINATIONS VIDEO

Adam Blokus, Adam Brzeski, Jan Cychnerski[1]

[1]Department of Computer Architecture, Gdańsk Technical University, Poland
ablokus@eti.pg.gda.pl, brzeski@eti.pg.gda.pl,
jan.cychnerski@eti.pg.gda.pl

ABSTRACT

The article presents a novel approach to medical video data analysis and recognition of bleedings. Emphasis has been put on adapting pre-existing algorithms dedicated to the detection of bleedings for real-time usage in a medical doctor's office during an endoscopic examination. A real-time system for analyzing endoscopic videos has been designed according to the most significant requirements of medical doctors. The main goal of the performed research was to establish the possibility of ensuring the necessary performance of a given class of algorithms to introduce the solution into real life diagnostics.

The structures of two exemplary algorithms for bleeding detection have been analyzed to distinguish and discuss parallelization options. After applying them to the algorithms, the usage of a supercomputer multimedia processing platform allowed to acquire the throughput and latency values required for real-time usage. Different configurations of the algorithms have been tested and their measured parameters have been provided and discussed.

KEYWORDS

Medical image classification, parallel algorithms, super computing

1. INTRODUCTION

Today, medical image processing tasks are not only getting larger but they are also getting more complicated as we try to analyze various processes with a constantly increasing accuracy. One of the directions the research in this field has taken in the recent years is the development of specialized recognition algorithms for supporting medical doctors in making diagnostic decisions. An important part of this direction of research involves the automatic recognition of images of the gastrointestinal tract by specialized classifiers and means of artificial intelligence. When such algorithm analyzes the video, the detected frames are presented to a medical doctor, who is responsible for their final classification. There are various works that relate to the problem of automated disease detection [1]. Many implemented algorithms perform remarkably well, achieving very high rates of accuracy. The developed algorithms include both general ones, adaptable to distinguish between different kinds of diseases, and specific ones, which are based on particular features of a disease.

Bleeding detection algorithms are one of the most often developed and analyzed classes of more specific image classification algorithms ([2, 3, 4, 5]). They find surprisingly many uses, as various kinds of dangerous lesions (e.g. cancers or ulcers) are often also a cause of gastrointestinal bleedings. Research of such algorithms focuses mainly on improving their

DOI : 10.5121/ijdps.2013.4401

accuracy, what leads to very promising results (e.g. an over 90% accuracy in [3]), on the other hand, though, the problem of the algorithms' efficiency is rarely considered. To utilize the possibilities given by automatic recognition, we undertook measures to adapt two of such algorithms for a real-time system, that could aid a medical doctor during an endoscopic examination. We have chosen three methods of parallelization, that can be applied easily on video processing services on our computational platform KASKADA (described in section 3) and can assure the necessary efficiency of recognizing algorithms.

Gastrointestinal bleeding detection algorithms are actually a wide class of diverse methods, adapted to detect bleedings visible on video frames. At the same time, they cannot be prone to mistaking bleeding with healthy tissue having a color close to red. For our research we have chosen two distinct algorithms to present how our methods of parallelization can be applied to different cases.

Our team has put work into introducing these algorithms into the traditional endoscopy, by adapting them to be useful not only in offline processing but also in the course of endoscopic examinations. While performing his duties, the doctor could be supported by a computer system giving hints and marking bleeding regions in the video image. Such system could vastly improve not only the diagnosis and disease detection rate, but also limit the time the doctor has to spend on documenting his findings. The most significant prerequisite for such system is high efficiency, so that it would not slow down the examination procedure.

Our aim was to create a system capable of performing real-time bleeding recognition in video data coming directly from the endoscope. To achieve this we have established the values of the throughput and the latency that the system must comply with. In the considered case, the throughput should be equal to at least 30 fps. It corresponds to the frame rate of the video provided by endoscopes (25 fps) with an additional overhead to ensure fluent processing. The latency of the recognition system is the average amount of time that passes between acquiring a frame and presenting the result of its classification to the doctor. Medical specialists cooperating with our project indicated that for an online analysis to be helpful during examinations, the latency parameter should fall within the range of 2 seconds. Such amount of time typically corresponds to the movement of the endoscope by not more than 5 cm in the gastrointestinal tract. After including the unavoidable delay of transferring data back and forth, about 0.2 seconds are left as a safe time margin for the video processing and recognition algorithms. Of course, from the medical specialist's point of view the overall latency of the system should be kept as low as possible.

Although systems that divide the data among computational nodes, like the one presented in [6], or analyze only a subset of video frames can increase the throughput, the latency value is still going to be too high if the algorithm analyzing a single frame proves to be too slow. Also, in a continuous processing of a video stream analyzing only selected frames would not be desirable. Therefore, the key for reducing the system's latency is to parallelize the processing algorithm itself.

A common method of increasing the throughput of algorithms is dividing them into logically consistent blocks, which are executed on separate nodes and process incoming frames as a pipeline. Most of the workflow-building solutions that can be found in literature, notably ones such as Pegasus [7], Triana [8], Taverna [9] and Kepler [10], operate in a grid environment. However, computations in a heterogeneous environment can often prove to be non-deterministic and not always reliable. Furthermore, grid nodes are usually connected by a slow, high-latency network that can hinder the whole system's effectiveness. Hence, we propose the use of a uniform and reliable cluster computer environment, dedicated for processing video.

Our research, which shows the capability of real-time endoscopic video stream processing, has been conducted on the multimedia stream processing platform called KASKADA [11], which is described in section 3. The platform is deployed on a cluster supercomputer environment which consists of multiple computational nodes connected by a fast low-latency network. It is capable of processing incoming data with the use of services created by the user and provides methods for an easy parallelization of the algorithms. During previous experiments on general classifying algorithms [12] complying to a similar scheme, the incorporation of the KASKADA platform into the system has allowed it to achieve the performance required for real-time usability.

During the initial tests, both chosen bleeding detection algorithms have proven to be too slow to satisfy the parameters imposed by real-time processing. Therefore we have subjected them to parallelization with the use of methods provided by the KASKADA platform. Afterwards, the possibility of real-time processing was tested by measuring latency and throughput of the endoscopic video analysis. The tests have shown that all prerequisites for the real-time processing can be met using the chosen methods.

The rest of the paper is organized as follows. In section 2 purely sequential versions of the two chosen algorithms are introduced. Section 3 contains the characterization of the execution environment of the KASKADA platform and its capabilities. Next, in section 4, general methods and modifications aimed towards parallelizing video analyzing algorithms are presented. The results are presented and discussed in section 5. Finally, section 6 encompasses conclusions and suggestions for future research.

2. THE SEQUENTIAL ALGORITHMS

From the set of published developed bleeding detection algorithms with a researched and established high quality of recognition (e.g. [2, 3, 13]), two different algorithms have been chosen for our analysis. The sequential versions of both of them have proven to be reliable enough to consider using them in a medical doctors office, if only their efficiency was high enough. In this section the details of both algorithms are presented, what allows to distinguish parallelizable parts and give a perspective on how speed improvements can be made.

Algorithm A [2] is based on transforming the image to a new color-space (HSV) and filtering out separate pixels of a frame in accordance with given ranges of color-space coordinates specific for blood and tissue. Afterwards, the image is processed in a series of steps to single out just the regions most suspected to be representing bleedings. On the other hand, algorithm B [3] incorporates more advanced methods of artificial intelligence. After dividing the incoming frame into a grid of smaller images, it describes each of them with a set of features (statistics of the particular piece of the grid) and utilizes an artificial neural network to classify them as either representing bleedings or not.

2.1. Algorithm A

The first of the tested algorithms consists of several steps that incrementally decrease the area of the image suspected of representing bleeding tissue. The algorithm has been outlined in figure 1A. Each block presents one logically separate step of the algorithm.

The consecutive steps of the algorithm are:

1. Dark pixel removal - eliminates from further analysis those pixels that are too dark to possibly contain blood or to be a recognizable part of the video (e.g. shadows).
2. Red pixel detection - identifies pixels that have various shades of the red color.

3. Edge masking - because specific characteristics of visible edges can lead to confusing them with some abnormalities, this step is responsible for masking the edges out.
4. In the next logical step two operations are performed in parallel and their result is combined afterwards:

   (a) Blood red pixel detection - finding pixels in a narrower spectrum of red, which can be classified as blood rather than as reddish tissue.
   (b) Anomaly detection - the goal of the anomaly detection subroutine is to find regions that are visually standing out from their background. The result is a mask of areas that are found to be differing from the statistical features of the whole frame.

5. Morphological operations - the combined masks from the previous step are morphologically eroded to present only solid and compact areas of a significant size.

All pixel discriminating steps (1, 2 and 4a) are simply defined by specifying the sought ranges of coordinates in the HSV color space.

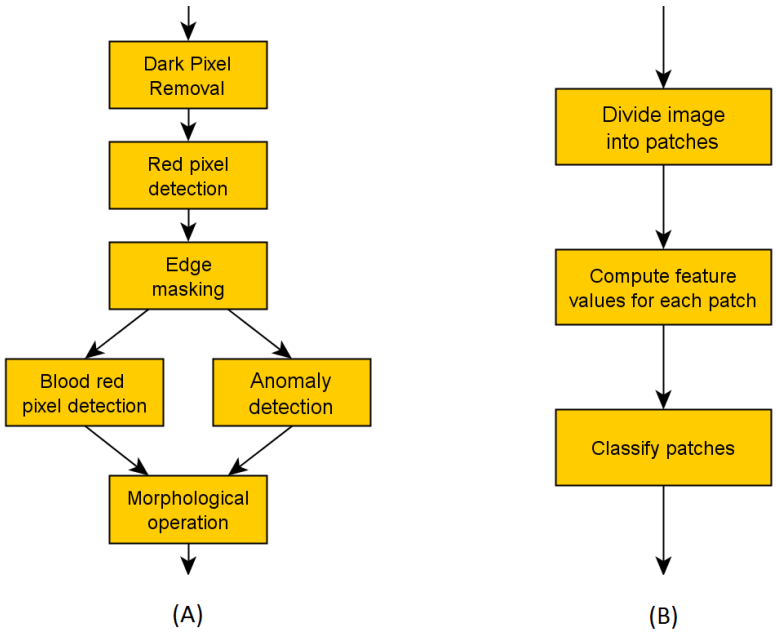

Figure 1: Logical block structure of algorithms A and B

One important feature of the algorithm is that it allows for an easy parallelization of its execution by handling each frame separately, as the algorithm does not take information from previous frames into consideration. The algorithm also requires performing many operations on the whole bitmap of one frame, what makes it difficult to parallelize it by distributing separate parts of a frame among computation nodes.

2.2. Algorithm B

The second algorithm that has been researched [3] takes a significantly different approach, as figure 1B presents.

In the first step of the algorithm, the frame is divided into a grid of 23×23 pixel patches - during the training and testing of this method we established this size to result with the best accuracy, while being only slightly smaller than the originally suggested 30×30 pixels. The second step, feature acquisition, is responsible for computing a set of statistics for each patch, which will be further used for the recognition. In the last step, a trained neural network classifier discriminates the patches representing bleeding tissue from other ones.

This algorithm reveals two important features which make it susceptible to parallelization. Primarily, every frame can be processed separately. The only data necessary to find bleedings on the image are the feature values of the patches from the current frame. The second advantage of this algorithm is that already by design different parts of the image are processed independently. Dividing the frame image into patches allows for distributing the processing of a single frame among multiple processing nodes that can handle separate parts of the image.

3. PRIMARY EXECUTION ENVIRONMENT

All experiments have been carried out in the environment provided by the multimedia processing platform KASKADA (Polish acronym for: *Contextual Analysis of Data Streams from Video Cameras for Alarm Defining Applications*), which was developed as a part of the MAYDAY EURO 2012 project at the Gdańsk University of Technology.

The KASKADA platform is an execution environment for algorithms which process multimedia streams and is deployed on the cluster supercomputer Galera Plus [14] (192 nodes with two Intel Xeon Six-Core 2.27 GHz processors, at least 16GB RAM, InfiniBand network). The main execution units on the KASKADA platform are the processing services. The first type of services are simple services, which can incorporate chosen algorithms. They can realize a particular task such as image conversion, or perform more complicated ones like a complete face detection algorithm.

Simple services can be organized into more sophisticated complex services forming a parallel workflow system. The platform is responsible for managing the life cycle of all the services, the connections between them and the input streams from various data sources, primarily - video streams. The incoming streams are decoded and given as input for the services. Each simple service can create new data streams, which can become the inputs for other simple services or the outputs of the whole complex service. Computations in the simple services are performed in parallel, as each service can be bound to different nodes of the supercomputer. This feature of the KASKADA framework encourages the design of complex algorithms in a fashion that allows to parallelize their execution by introducing logically separate blocks and pipeline processing of the incoming data.

The services are also capable of signaling detected contextual events (a dangerous situation, a disease, etc.). Those events are sent to a specified queue server and can be read by an authorized, external application.

The structure of the platform makes it reasonable to perceive it as a possible component of complex diagnostic systems that can aid medical doctors with real-time analysis of video data from endoscopic examinations. Thus, all our efforts to improve the algorithms are focused on its parallelization with the tools provided by the KASKADA platform.

4 PARALLELIZATION OPTIONS

Initial speed tests of the sequential algorithm (presented in section 5) have shown, that neither of the two selected algorithms can be directly used in real-time video analysis. They have been proven to be unable to process the incoming video stream in real-time due to the insufficient throughput values. This also made the analysis of the algorithms' latencies irrelevant, as due to the insufficient throughput the algorithm would keep falling behind more and more in the course of the examination.

During previously performed experiments [12], the logical structure of the analyzed algorithm has been utilized for the parallelization in a pipeline-like scheme. In this experiments we present another two methods, applicable to a wider range of algorithms. The first of the presented method allows for the use of any kind of a black-box algorithm classifying single frames. On the other hand, the second one is based on some assumptions about the reasoning process.

Basing on the possibilities given by the KASKADA platform we have established some universal methods of parallelization which could be applied on the algorithms. They are aimed both at some general properties of video processing algorithms (distributing the processing of various frames and images onto various nodes) and specific, but quite common, properties of the algorithms (e.g. possibility to process parts of a frame separately). Below we present the adopted methods of parallelization.

4.1. Distributing frames among nodes

First experiments were performed solely with the use of parallelization methods provided by the KASKADA platform. Service startup, data flow, event message passing and, most notably, computation nodes allocation were all maintained by the execution environment of the platform. The platform also ensures that all processing blocks are given the predefined computation resources. Those mechanisms guaranteed that each computing service block used during the experiments had an exclusive access to at least one processing core.

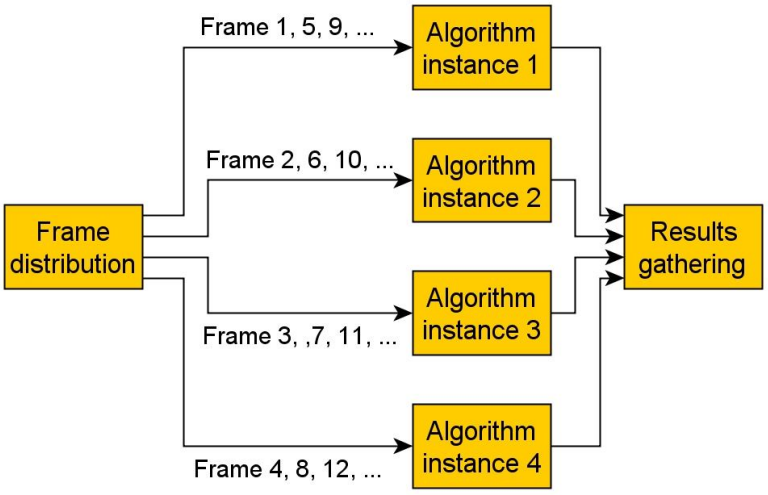

Figure 2: Cyclic distribution of frames among four computation nodes

This method of parallelization involved introducing multiple instances of the algorithm in the form of simple services on separate nodes. The incoming frames are distributed in a cyclic manner among consecutive nodes, as presented on figure 2.

This method is limited only by the number of available computational nodes and can be used to increase the processing throughput efficiently. In offline processing an arbitrary number of nodes can be utilized. For processing the video stream in real-time, we can approach a boundary value for the number of computational nodes, above which the system starts experiencing idle time with computational nodes awaiting for incoming frames.

4.2 Data decomposition

As it was shown in section 2, some algorithms can be parallelized by processing different parts of a frame on separate nodes, but only as long as their reasoning is based on separate pieces of the whole image. Nevertheless, the data decomposition method is also susceptible to being adopted for a broader class of algorithms by the introduction of an overlap. Dividing the image into overlapping patches allows to process each patch separately with data about his closest neighborhood taken into consideration, what might be just enough for some algorithms. This method is presented on figure 3, where we can see an exemplary distribution of computations on a single frame among four computation nodes. This method suits algorithm B especially well, because it conforms to the division of the frame that the algorithm performs in its first step.

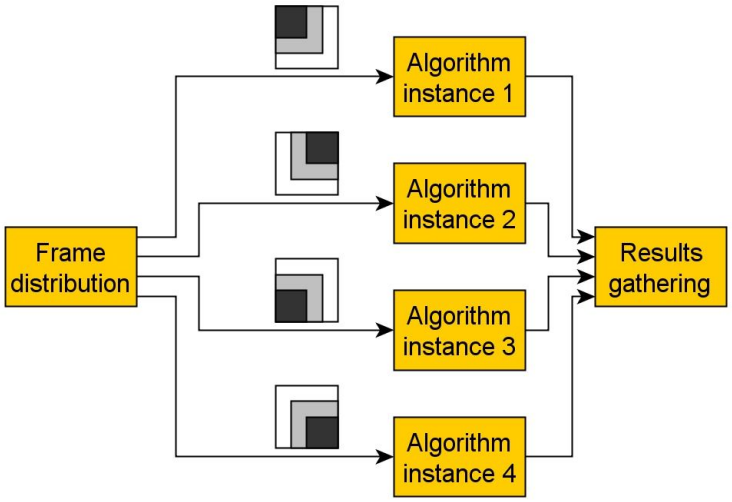

Figure 3: Distribution of a frame (white contour) split into four pieces (black) with overlaps (gray)

In cases such as that of algorithm A, introducing this method requires a deeper analysis of how (and whether) it influences the classification results and finding a reasonable trade-off between the performance improvement and the possible loss of accuracy. We have assumed that a 5% discrepancy between the classification results of the whole image and combined results of its subparts is still acceptable.

4.3 Additional parallelization

To make further acceleration improvements we also decided to parallelize calculations in the simple services. Each simple service deployed on the KASKADA platform is assigned to a

multiprocessor cluster node of the supercomputer at run-time. In consequence, classic process parallelization techniques such as dividing computations into multiple threads can be efficiently employed. To achieve this, we chose an OpenMP library which allows smooth parallelization of the most computationally intensive parts of code, such as loops.

5 TEST RESULTS

All experiments have been carried out with different modifications of the algorithms, as described in the previous section. The results have been summarized in Tables 1 and 2 which show the average values of the chosen measures from all performed tests. More detailed results of the throughput measurement have been presented in Figures 4 and 6 which shows the outcome of each test.

The testing procedure was the same for each algorithm and it proceeded as follows: The first tests involved the sequential version of the algorithm. In both cases we confirmed its poor performance for live video processing. Therefore, next steps have been taken to parallelize the algorithm.

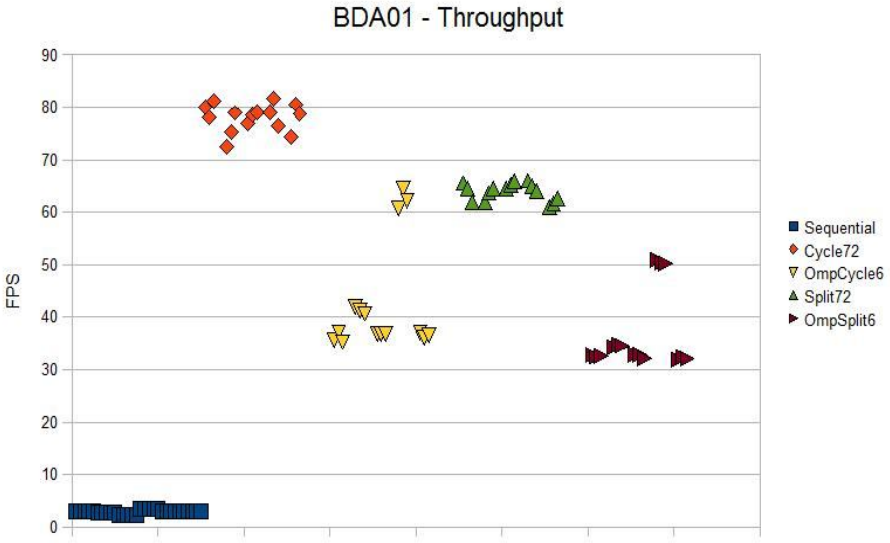

Figure 4: Diagram of the throughput(H) values acquired in the tests of algorithm A

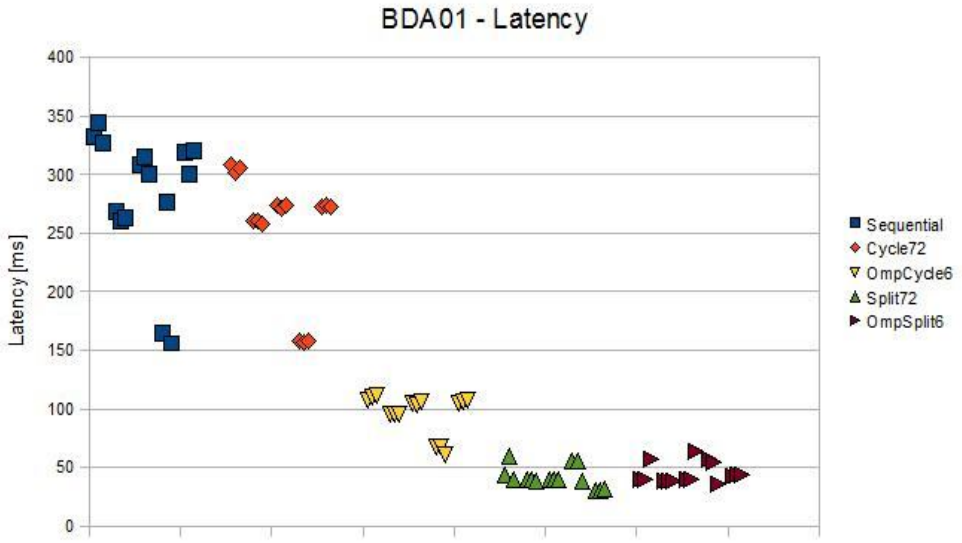

Figure 5: Diagram of the latency(L) values acquired in the tests of algorithm A

Each of the algorithms was tested both with the cyclic frame distribution modification and, separately, with the frame-splitting modification. The next step involved utilizing all available computational power of the multi-core nodes by introducing further parallelization with the OpenMP library.

Additional measurements were performed for measuring the effect of division into blocks on the recognition quality of algorithm A. The frame-splitting modification of algorithm A still performed well after dividing the incoming frames into 6 or 72 sub-pictures, even after reducing the overlap size to 0. This allowed the final performance tests to be carried out without overlaps.
Figures 5 and 7 present the latency measurement results of the tested algorithms. Latencies of different runs for particular test cases have been shown grouped together. As we can observe, for most cases the latency is kept far below the upper boundary. In the case of applying only the circular frame distribution method, the latency values have been slightly increased in comparison to the sequential version, probably due to additional communication costs. Although this is the only case where the average latency exceeds the established limits, after assuring a high speed network connection between the examination room and the supercomputer even such algorithm could be used in real-life applications.

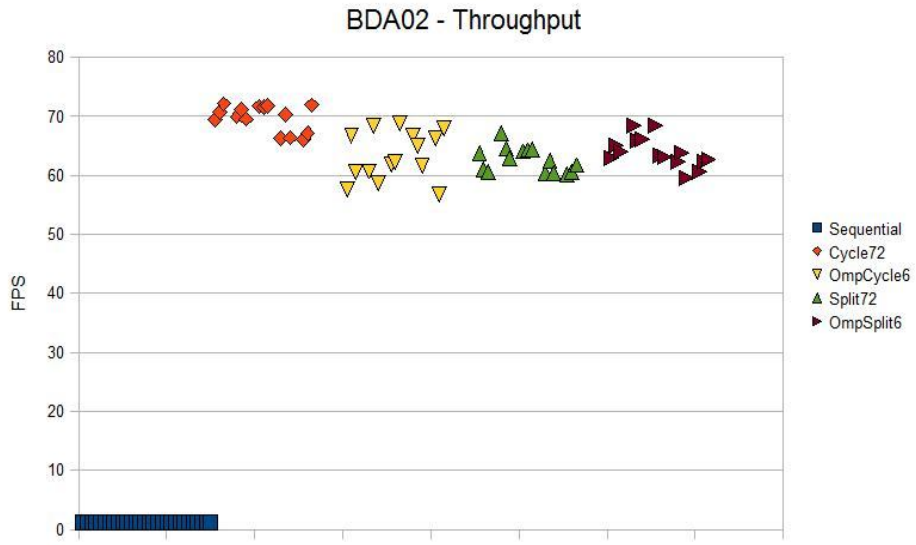

Figure 6: Diagram of the throughput(H) values acquired in the tests of algorithm B

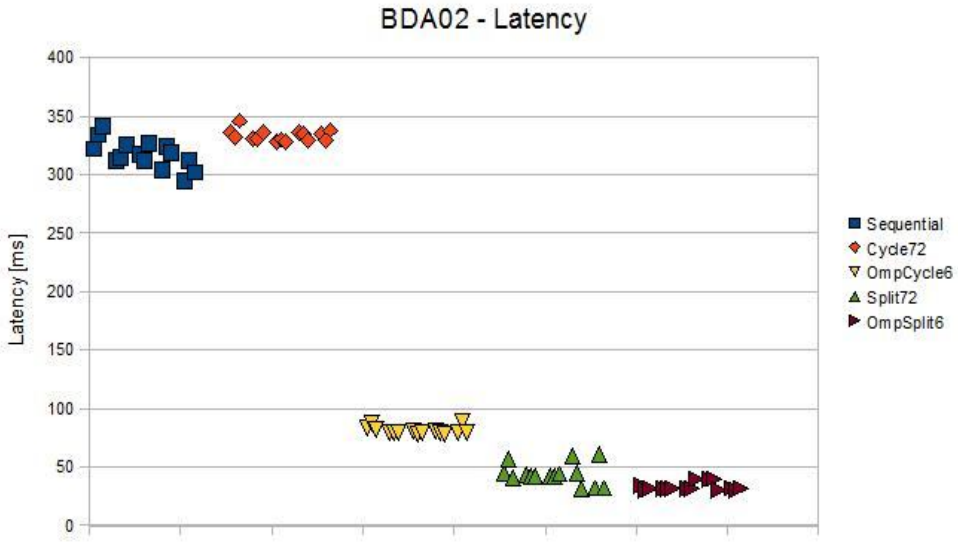

Figure 7: Diagram of the latency(L) values acquired in the tests of algorithm B

As it can be seen in the summaries in Tables 1 and 2, the parameters of the final algorithms fit into the established boundaries. This holds for both tested algorithms and all methods of parallelization.

Table 1: Summarized results of the performed experiments for algorithm A

Algorithm Version	H[fps] (throughput)	σ_H	L[ms] (latency)	σ_L
Sequential	2.48	1.22	-	-
Cyclic frame distribution (72 nodes)	79.76	1.59	305.67	3.51
Cyclic frame distribution (6 nodes) + OpenMP (12 threads)	42.64	10.51	96.13	16.9
Frame split (72 nodes)	63.88	1.66	41.27	8.96
Frame split (6 nodes) + OpenMP (12 threads)	36.47	7.34	44.6	8.86

Table 2: Summarized results of the performed experiments for algorithm B

Algorithm Version	H[fps] (throughput)	σ_H	L[ms] (latency)	σ_L
Sequential	1.72	0.1	-	-
Cyclic frame distribution (72 nodes)	69.75	2.2	333.33	4.84
Cyclic frame distribution (6 nodes) + OpenMP (12 threads)	63.32	4.07	81	3.18
Frame split (72 nodes)	62.56	2.08	43.87	9.13
Frame split (6 nodes) + OpenMP (12 threads)	63.87	2.51	32.8	3.53

6 CONCLUSION

The performed experiments have proven that real-time recognition of bleedings in video from endoscopic examinations is possible with a proper choice of hardware (a parallel multimedia processing platform) able to parallelize various kinds of available algorithms. Although these prerequisites are quite extensive and demanding, the acquired results are valuable as one of the first in this direction of research, as most other experiments focus purely on the accuracy of the off-line processing algorithms.

The determined value of the system's processing latency is relatively low in comparison with the boundary value. Therefore, in order to provide a satisfying performance of the complete system the Internet connection is allowed to introduce a latency of not more than 1.9s - a value not difficult to achieve.

All of the results were made possible by the use of the supercomputer multimedia processing platform KASKADA. It allowed an easy and straightforward parallelization of the chosen algorithms and provided the runtime environment for efficient video processing. With just one exception, the applied methods do not influence the reasoning and recognition process of the algorithms. Therefore, the parallelization does not influence the reported high accuracy of the algorithms. Other algorithms (e.g. [4,5]), can be parallelized in an analogous manner. Further research in this direction is planned to involve a real life installation of a diagnosis support system at the Medical University of Gdańsk to evaluate its usability.

ACKNOWLEDGMENT

Research funded within the project No. POIG.02.03.03-00-008/08, entitled "MAYDAY EURO 2012 - the supercomputer platform of context-depended analysis of multimedia data streams for identifying specified objects or safety threads". The project is subsidized by the European regional development fund and by the Polish State budget.

REFERENCES

[1] A. Karargyris and N. Bourbakis, "Wireless capsule endoscopy and endoscopic imaging: A survey on various methodologies presented," IEEE Engineering in Medicine and Biology Magazine, vol. 29, no. 1, pp. 72–83, 2010.

[2] B. Penna, T. Tillo, M. Grangetto, and E. Magli, "A technique for blood detection in wireless capsule endoscopy images," 2009 European Signal, no. Eusipco, pp. 1864–1868, 2009.

[3] B. L. B. Li and M. Q. H. Meng, "Computer-aided detection of bleeding regions for capsule endoscopy images.," IEEE Transactions on Biomedical Engineering, vol. 56, no. 4, pp. 1032–1039, 2009.

[4] A. Karargyris and N. Bourbakis, "A methodology for detecting blood-based abnormalities in wireless capsule endoscopy videos," 2008 8th IEEE International Conference on BioInformatics and BioEngineering, pp. 1–6, Oct 2008.

[5] Y. S. Jung, Y. H. Kim, D. H. Lee, and J. H. Kim, "Active blood detection in a high resolution capsule endoscopy using color spectrum transformation," 2008 International Conference on BioMedical Engineering and Informatics, pp. 859–862, May 2008.

[6] K. Ioannis, S. Tsevas, I. Maglogiannis, and D. Iakovidis, "Enabling distributed summarization of wireless capsule endoscopy video," in Imaging Systems and Techniques (IST), 2010 IEEE International Conference on, pp. 17 –21, july 2010.

[7] E. Deelman, J. Blythe, A. Gil, C. Kesselman, G. Mehta, S. Patil, M. hui Su, K. Vahi, and M. Livny, "Pegasus: Mapping scientific workflows onto the grid," pp. 11–20, 2004.

[8] S. Majithia, M. Shields, I. Taylor, and I. Wang, "Triana: A graphical web service composition and execution toolkit," Web Services, IEEE International Conference on WebServices, vol. 0, p. 514, 2004.

[9] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," Bioinformatics, vol. 20, no. 17, pp. 3045–3054, 2004.

[10] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the kepler system: Research articles," Concurr. Comput. : Pract. Exper., vol. 18, pp. 1039–1065, 2006.

[11] H. Krawczyk and J. Proficz, "Kaskada – multimedia processing platform architecture," SIGMAP, 2010.

[12] A. Blokus, A. Brzeski, J. Cychnerski, T. Dziubich, and M. Jędrzejewski, Real-Time Gastrointestinal Tract Video Analysis on a Cluster Supercomputer, p. 55–68. Springer, 2012.

[13] A. Brzeski, Parameters optimization in medicine supporting image recognition algorithms. 2011.

[14] http://top500.org, "Top 500 supercomputer sites," 2012.