Execution Time Improvement using CPU Parallelization and Non-Uniform High-Resolution Range-Doppler Map Estimation in HFSWR

Dragan Golubović, Nenad Vukmirović, Zoran Lončarević, Marko Marković and Miljko Erić

Abstract—**High-resolution range-Doppler (RD-HR) map estimation, used for primary signal processing in a High Frequency Surface Wave Radar (HFSWR), is the most computationally demanding step of the vessel detection algorithm. In order to reach real-time processing, which is of great importance in practical implementations of such systems, a very high-speed computation is required. In this paper, we propose non-uniform signal frame selection, to reduce the load with almost no loss in performance, and parallel processing on a CPU to get high-resolution range-Doppler maps in a multi-antenna scenario. The paper contains the description of the proposed algorithm and the performance analysis. The experimental results show a 60- to 130-fold improvement in the execution time of the program for vessel detection.**

Index Terms—**HFSWR; range-Doppler map; high-resolution methods; multi-core; parallelized algorithms; speedup.**

I. INTRODUCTION

The focus of numerous scientific papers is maritime surveillance of vessels at long distances. The reason for that are many illegal crime activities over the horizon, drug trafficking, attacks on petrol platforms and strategic objects, etc. High Frequency Surface Wave Radars (HFSWRs) are widely used for this purpose [1]-[3].

To make these systems have better ship detection accuracy, as well as the ability to detect some ships, which are not visible at all using the currently used primary signal processing algorithms, high-resolution algorithms are used [4].

The downside is that the computational burden of HFSWRs is huge in that case, and high-speed computation is required in order to have real-time processing. It is of great importance in practical implementations.

In recent years, the demand for high performance numerical computing in many radar systems was increased. The focus of many papers is the formulation of parallel algorithms for the MIMO radar based on the Central Processing Unit (CPU)/ Graphical Processor Unit (GPU) architecture [5]-[9]. The architecture must be capable of multitasking, which allows multi-threaded execution of the program for vessel detection. GPUs have an essential role in designing the real-time programs because they are highly parallel, multithreaded processors. Graphic cards are equipped with multi-core GPUs enabling the development of high computationally demanding programs. But the performance improvement of multithreaded programs depends on the algorithms used and their implementations, which is often not a trivial task. This improvement is limited by the fraction of the program source code that can be run on multiple cores simultaneously. It is important to say that high execution efficiency is possible, only if hardware characteristics are appropriate.

The focus of this paper will be a real-time implementation of target detection in HFSWRs by improving the existing implementation of the high-resolution range-Doppler (RD-HR) map estimation, which is the most computationally demanding task of the entire algorithm. The algorithm runs in real-time if its execution time is shorter than the acquisition time of the frames used for the estimation (32.768 s in this particular case).

The existing implementation did not run in real-time, so we solved this problem in two ways. Firstly, we significantly reduced the computational load of the RD-HR map estimation with very little performance loss by using non-uniform sampling across the frames. Secondly, the parallel signal processing on CPU was performed. An advantage of the proposed implementation is that it requires only a general-purpose computer and no expensive dedicated hardware, like graphic cards, or Digital Signal Processing (DSP) cards.

The paper is organized as follows. In Section II, we presented a detailed algorithm description to estimate high-resolution range-Doppler map. In Section III, we explained the numerically efficient method to achieve real-time processing requirement. We discussed some experimental results and made some comparisons in Section IV and

Dragan Golubović is with the University of Belgrade, School of Electrical Engineering, 11120 Belgrade, Serbia and Vlatacom Institute, 11070 Belgrade, Serbia (e-mail: dragan.golubovic@vlatacom.com).

Nenad Vukmirović is with the University of Belgrade, School of Electrical Engineering, 11120 Belgrade, Serbia and the Innovation Center of the School of Electrical Engineering, 11120 Belgrade, Serbia.

Zoran Lončarević is with Vlatacom Institute, 11070 Belgrade, Serbia (e-mail: zoran.loncarevic@vlatacom.com).

Marko Marković is with Vlatacom Institute, 11070 Belgrade, Serbia (e-mail: marko.markovic@vlatacom.com).

Miljko Erić is with Vlatacom Institute, 11070 Belgrade, Serbia and the University of Belgrade, School of Electrical Engineering, 11120 Belgrade, Serbia (e-mail: miljko.eric@vlatacom.com).

outlined some conclusions and further research activities in Section V.

II. HIGH-RESOLUTION RANGE-DOPPLER MAP ESTIMATION

HFSWR to be analyzed in this paper operates in HF spectrum 3-30 MHz and it uses Frequency Modulated Continuous Waves (FMCW). The radar consists of a transmitter (Tx) antenna array, receiver (Rx) antenna array and transceiver hardware. The numerical results in this paper are based on the signals from an N-element linear Rx antenna array. Other geometries are also possible. At the receiving channels, acquired complex time samples of IQ branch signals are available for further signal processing. In Fig. 1 the complete high-resolution target detection algorithm in HFSWR was proposed.

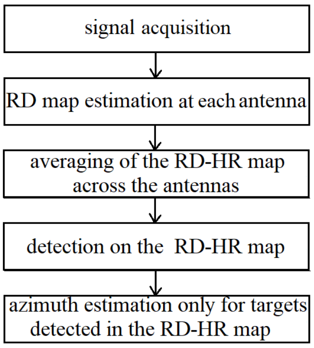


Fig. 1. The overview of the proposed high-resolution algorithm for vessel detection in HFSWR

The proposed algorithm has five steps, but the most computationally demanding task is step two – RD-HR map estimation at each antenna. Because of that, in order to improve program performance in terms of execution time, it is necessary to describe the creation of RD-HR map in detail and find a way to reduce its numerical complexity. So, other steps are not the focus of this section and only RD-HR creation will be explained.

In practical situations, P (the number of signal samples in one frame) and N are predefined values and M can be varied and it represents the number of frames used for the creation of one segment. Based on this segment, RD-HR map is formed. The developed algorithms were tested for the length of the segment of M=256, where the successive segments overlap in 128 frames. This ensures that the results are refreshed every 128 frames. The first step in the RD-HR creation process, is the implementation of FFT algorithm in P points for all frames m=1, 2, ... , M and all antennas n=1, 2, ... , N. By adding zeros to the vectors with signal samples, better grid resolution can be achieved when applying FFT. For further processing and a better estimate of the covariance matrix later, we need to extend the segment by L-1 additional frames. Fig. 2 shows the forming of the first two segments, with a 128-frame overlap. The same procedure is used for all other segments during signal processing.

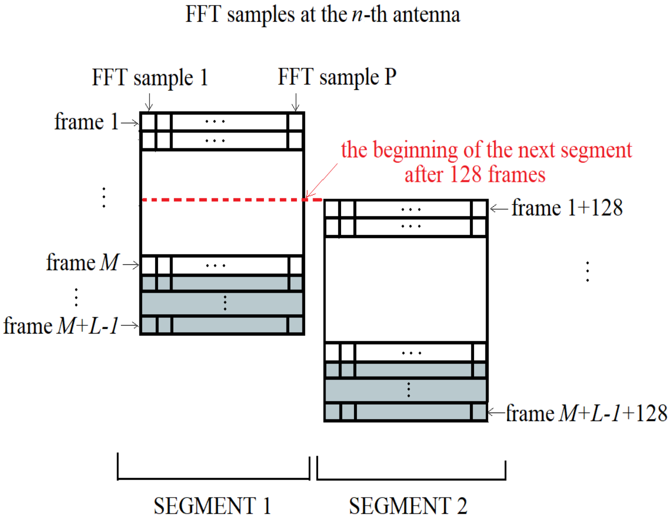


Fig. 2. Segment creation in the proposed algorithm

For each range cell, with index p, and each antenna, n, we form a matrix $\mathbf{Q}_{p,n} \in \mathrm{C}^{M\times L}$ as in Fig. 4 from the p-th FFT sample from each frame in Fig. 3. Columns of matrix $\mathbf{Q}_{p,n}$ are vectors $\mathbf{q}_{l,p,n}$, for $1 \le l \le L$, $1 \le p \le P$ and $1 \le n \le N$.

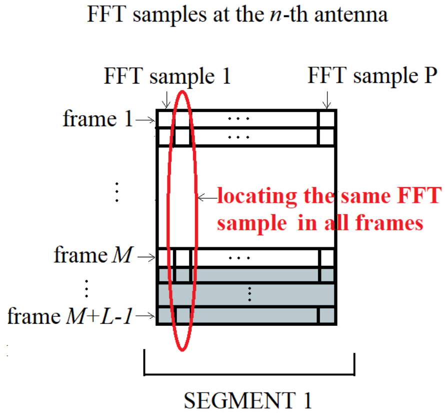


Fig. 3. Locating the same FFT sample in all frames

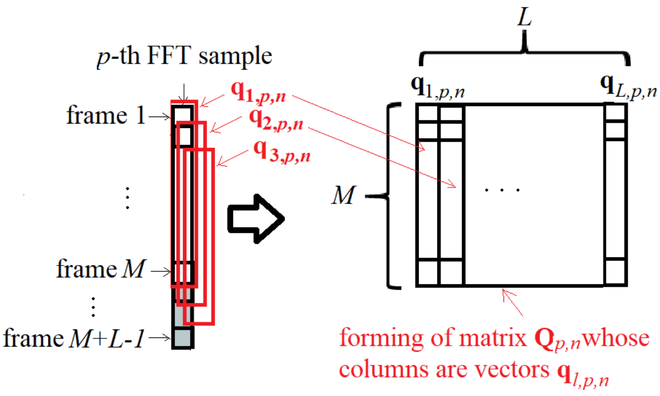


Fig. 4. The creation of the $\mathbf{Q}_{p,n}$ matrix

Then the covariance matrices $\mathbf{C}_{p,n} \in \mathrm{C}^{M\times M}$ are formed for n=1, 2, ... , N and p=$P-R$+1, $P-R$+2, ... , P, where R represents the maximum projected radar range, as follows:

$$\mathbf{C}_{p,n} = \frac{1}{L}\mathbf{Q}_{p,n}\mathbf{Q}_{p,n}^{\mathrm{H}} \in \mathrm{C}^{M\times M} \ . \qquad (1)$$

The formation of the covariance matrix $\mathbf{C}_{p,n}$ is a key step in the RD-HR map creation. Based on the covariance matrix, the criterion function of high-resolution MUSIC-based algorithm is formulated, as follows:

$$P_{\mathrm{MUS}}^{RD}(\mu, p, n) = \frac{1}{\left\|\mathbf{a}_{\mu}(\mu)^{\mathrm{H}}\,\mathbf{E}_{p,n}\right\|}, \qquad (2)$$

where the columns of $\mathbf{E}_{p,n} \in \mathrm{C}^{M\times(M-K)}$ are the M-K eigenvectors from the noise subspace of $\mathbf{C}_{p,n}$, corresponding to the M-K smallest eigenvalues of the covariance matrix $\mathbf{C}_{p,n}$, K is a parameter of the MUSIC-based algorithm, and $\mathbf{a}_{\mu}(\mu) \in \mathrm{C}^{M\times 1}$ is a steering vector formulated in the normalized Doppler domain as

$$\mathbf{a}_{\mu}(\mu) = \left[1, e^{-j\mu}, \dots , e^{-j\mu(M-1)}\right]^{\mathrm{T}} , \qquad (3)$$

where the parameter μ denotes the normalized Doppler frequency in radians per frame. The criterion functions are calculated for a set of discrete values of normalized Doppler frequencies and with a grid resolution that is many times better than the Doppler FFT resolution, thus obtaining an RD-HR map.

III. NUMERICALLY EFFICIENT METHOD FOR RD-HR MAP ESTIMATION

As presented in the previous section, the RD-HR map estimation is numerically complex (long execution time). Program for vessel detection of the proposed algorithm must be improved to achieve real-time requirement. Because of that, we propose here a numerically efficient method realized in two steps.

In the first step, we want to make some kind of pattern according to which we select a small subset of the frames from each segment. The reason for that is to reduce numerical complexity. In that case, the dimensionality of the covariance matrix will be smaller ($J \times J$) where $J < M$. The problem of non-uniform sampling method is analogous to the problem formulated in the field of antenna arrays - how to replace a linear uniform antenna array with a non-uniform antenna array with the same aperture and a smaller number of antennas without significant degradation of the antenna array factor. This problem in antenna array theory is known as the problem of minimally redundant linear antenna arrays. Fig. 5 shows the forming of the matrix $\mathbf{Q}^{(l)}_{p,n}$ by selecting a subset of J rows of the matrix and according to a chosen mapping $\ell : \{1, 2, \dots , J\} \rightarrow \{1, 2, \dots , M\}$, $J < M$.

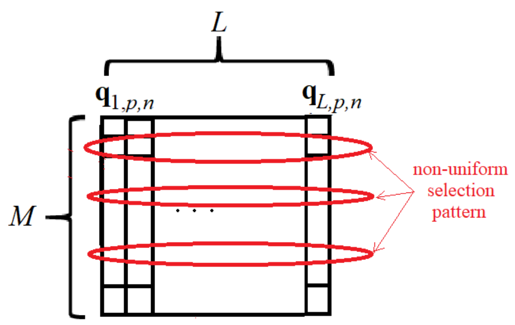


Fig. 5. The creation of the $\mathbf{Q}^{(l)}_{p,n}$ matrix using some non-uniform pattern

The same mapping ℓ is used to form the steering vector $\mathbf{a}_{\mu}^{(l)}(\mu)$ by non-uniform selection of the elements of the vector $\mathbf{a}_{\mu}(\mu)$.

The criterion function of the high-resolution MUSIC-type algorithm for creating RD-HR map with non-uniform sampling has the same form as the criterion function for the variant with uniform sampling, as follows:

$$P_{\mathrm{MUS}}^{RD\,(l)}(\mu, p, n) = \frac{1}{\left\|\mathbf{a}_{\mu}^{(l)}(\mu)^{\mathrm{H}}\ \mathbf{E}_{p,n}^{(l)}\right\|} \ . \qquad (4)$$

The numerical complexity is significantly reduced, because the eigenvalue decomposition of the covariance matrix is numerically simpler in this case.

Despite the significant reduction in execution time, real-time processing is still not achieved. Therefore, we perform another step in the algorithm optimization.

As can be seen, a similar signal processing is performed on each of the antennas. This leads us to the idea not to form RD-HR maps at all antennas sequentially, one after the other, but simultaneously. Therefore, it is necessary to create a multi-threaded process.

Fig. 6 shows the proposed method for multithreaded RD-HR estimation in order to reduce execution time. A thread is the smallest unit of a processing that can be scheduled by an operating system. The multithreading was realized on CPU cores.

Therefore, the most demanding job, which is the formation of RD maps, was realized through parallel execution on multiple cores, while the rest of the program, which is not computationally demanding, remained to be executed sequentially. The algorithm is also applicable to a system with more antennas, but the performance depends directly on the number of processor cores.

IV. NUMERICAL RESULTS

The results presented in this section are based on the measured radar data. P=1536 and L=64 are predefined values and the developed algorithm was tested for the length of the segment M=256, where the successive segments overlap with 128 frames. Number of antennas directly affect the execution time of the program for vessel detection, and we made some tests with N=16 antennas in the Rx antenna array.

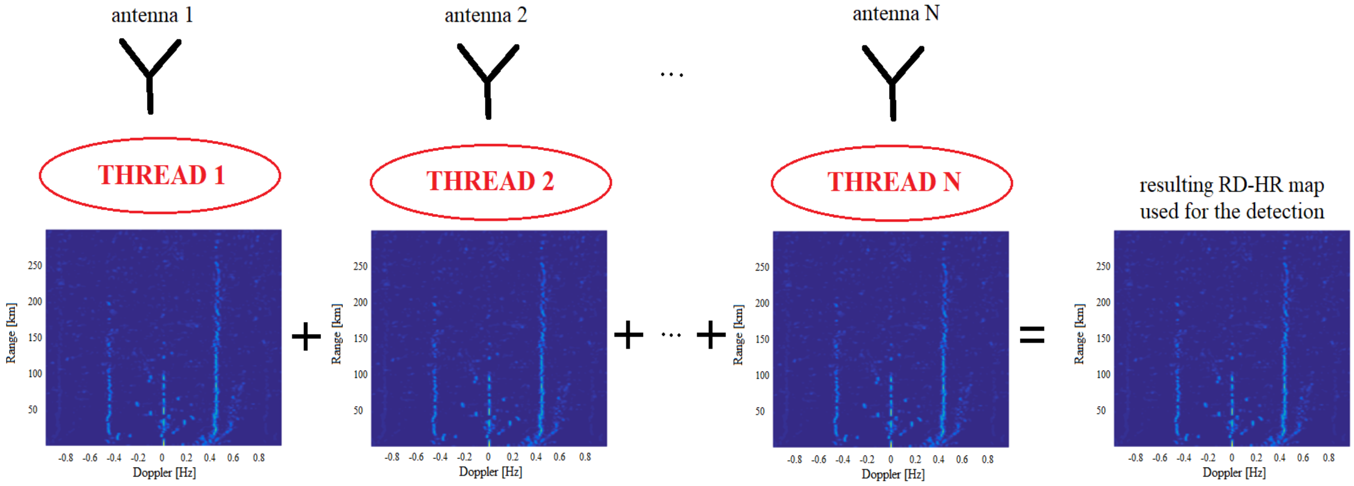


Fig. 6. Multithreaded RD-HR maps creation simultaneously at all antennas

The selected frame duration is 0.256 seconds, and since we want to output the results after every 128 frames, the real-time requirement is 32.768 seconds. Thus, the processing time of one segment must be shorter than 33.28 seconds. We choose also 3 predefined values for the parameter K of the MUSIC-based algorithm: K=10 for ranges up to 120 km, K=20 for ranges from 120 to 200 km, and K=30 for ranges up to 300 km. The number of grid points along the Doppler frequency dimension is 513. These are actually the basic system parameters.

For testing purposes, a program was developed using the programming language C and multithreading is realized too. We first formulate a testing methodology, seen in Fig. 7.

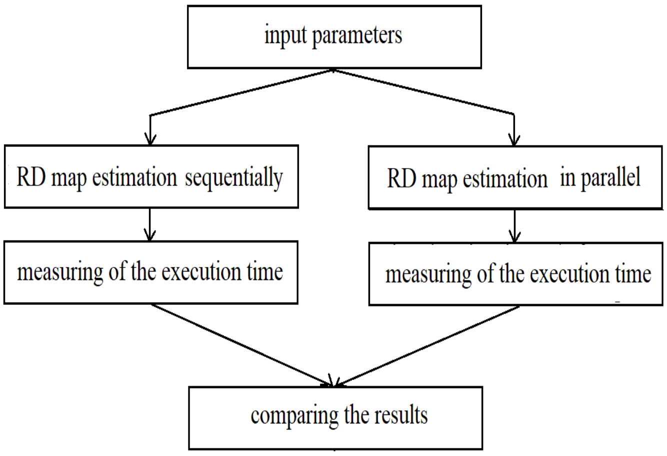


Fig. 7. The overview of the proposed testing methodology

We run the program on a PC with 6-cores CPU (i7), and on a PC with better 8-cores CPU (AMD Ryzen), and for 2 predefined patterns in order to get non-uniform RD-HR maps. The comparison with uniform RD-HR map is also presented. The first pattern is based on prime numbers between 1 and 256 (the last frame number in the segment) and the length of this pattern is J=56, as follows: pattern56={1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 256}.

The length of the second pattern is 88 and it is also based on prime numbers, with some more numbers inserted, in order to reduce the distance between the numbers in the pattern. This leads us to create an RD-HD map that visually resembles a uniform RD HR map, while being numerically simpler: pattern88 ={1, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 35, 37, 41, 45, 48, 50, 53, 55, 59, 61, 63, 67, 71, 73, 76, 77, 83, 85, 89, 92, 97, 101, 103, 105, 107, 109, 111, 113, 115, 118, 121, 125, 127, 129, 131, 134, 136, 137, 139, 142, 145, 147, 149, 151, 155, 157, 160, 163, 167, 171, 173, 179, 181, 184, 187, 191, 193, 197, 199, 202, 205, 209, 211, 215, 219, 223, 227, 229, 233, 236, 239, 241, 245, 247, 251, 254, 256}.

Fig. 8-10 show RD-HR maps for different patterns.

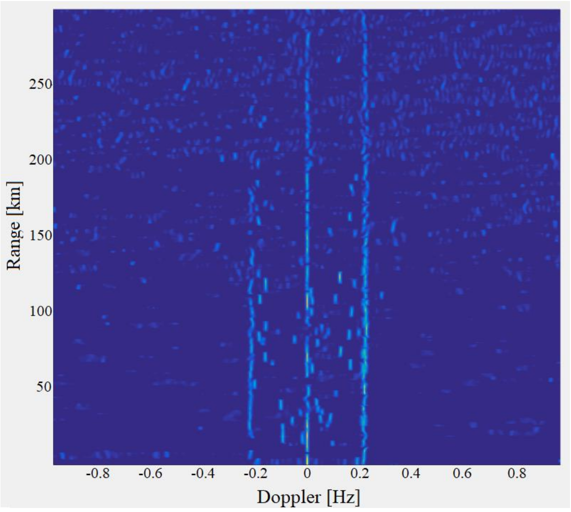


Fig. 8. Uniform obtained RD-HR map of the first segment

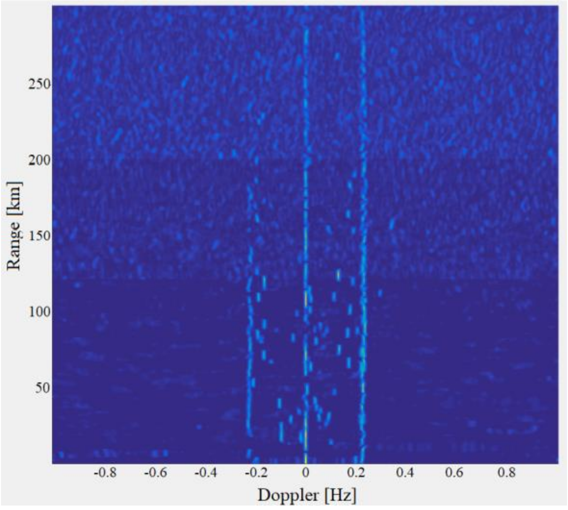


Fig. 9. Non-uniform obtained RD-HR map of the first segment (pattern88)

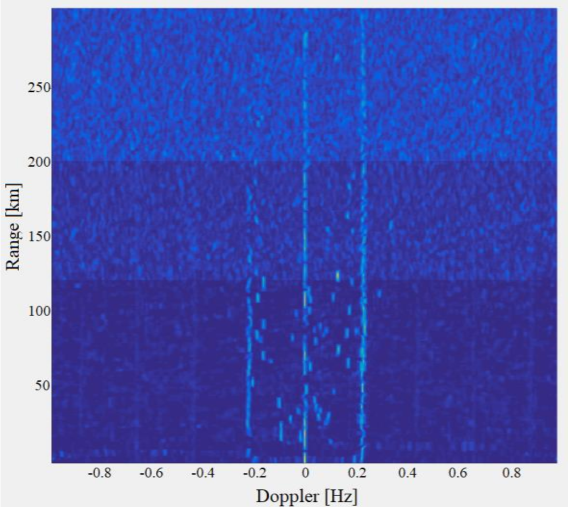


Fig. 10. Non-uniform obtained RD-HR map of the first segment (pattern56)

As can be seen from the figures, non-uniform sampling of frames increases the number of peaks in the RD-HR map compared to the case with uniform sampling, but vessels on the map are clearly visible in all 3 cases.

The dominant tasks in terms of the number of complex multiplications and additions are the eigenvalue decomposition and the calculation of MUSIC criterion function [10]. The approximate number of operations is

$$N_{\mathrm{op}} \sim \frac{13}{3} J^3 RN + N_{\mathrm{d}} J(J+1) RN \,. \tag{7}$$

In uniform variant $J=M$. Thus, we decrease this number by a factor 44.6 and 15.3 for pattern56 and pattern88, respectively. N_{d} is the length of RD-HR map by Doppler dimension. In this particular case $N_{\mathrm{d}}=513$. Since the numerical complexity is much lower, a non-uniform variant can be used for real-time processing.

Table I shows the execution time of the program (with no-parallelized code) for vessel detection on different computers. Also, the measured execution time to form the RD map is compared to the measured execution time of the rest of the program.

TABLE I

EXECUTION TIME OF THE PROGRAM (NO-PARALLELIZED CODE)

CPU type	**Execution time of RD-HR estimation (seconds)**	**Execution time of the rest of the program (seconds)**
Intel CORE i7 1075H	880	3.12
AMD Ryzen 9 5900HX	606	2.64

The results clearly show that the estimation of the RD-HR map is numerically most complex, and the execution time improvement is required in order to reach real-time processing.

Now, we can define the speedup S and the efficiency E by comparing the execution time on one core, T_1, and on n cores, T_{n}:

$$\mathrm{S} = \frac{T_1}{T_n} \tag{5}$$

$$E = \frac{T_1}{nT_n} \,. \tag{6}$$

Table II shows the comparison between the execution time of the program with and without parallelized code. The obtained results show that the real-time processing is achieved.

TABLE II

EXECUTION TIME OF THE PROGRAM (PARALLELIZED CODE)

CPU type and selected pattern	**Execution time of the program without parallelization (seconds)**	**Execution time of the program with parallelization (seconds)**
Intel CORE i7 1075H (uniform)	883.12	132
Intel CORE i7 1075H (pattern88)	91.2	14.21
Intel CORE i7 1075H (pattern56)	32.12	7.54
AMD Ryzen 9 5900HX (uniform)	608.64	90.75
AMD Ryzen 9 5900HX (pattern88)	60.32	9.63
AMD Ryzen 9 5900HX (pattern56)	24.56	4.4

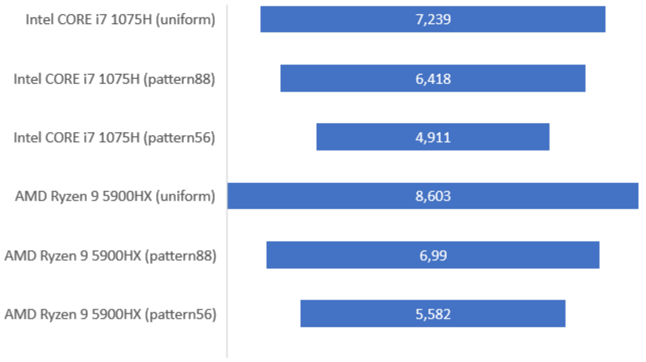


Fig. 11. Speedup of the parallelized algorithm

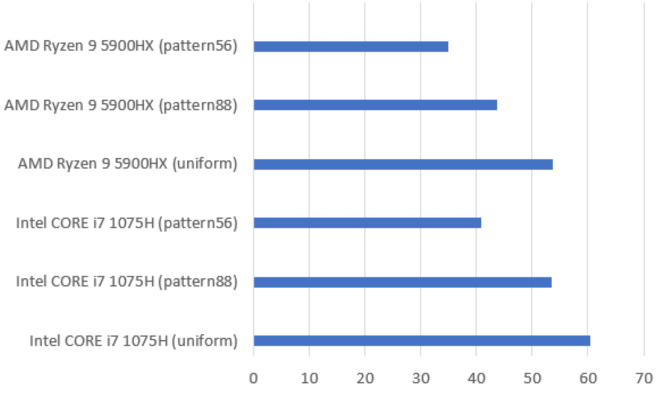
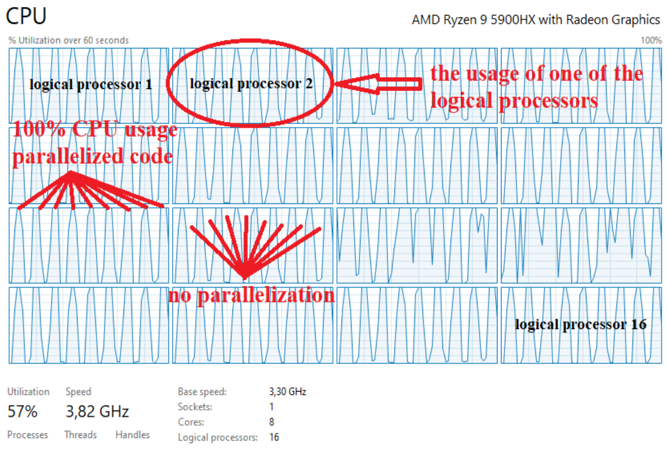


Fig. 12. Efficiency of the parallelized algorithm

Fig. 11 and 12 show speedup and the efficiency of the parallelized algorithm. Logical processors usage in multithread software was shown in Fig. 13.



Utilization
57%
Speed
3,82 GHz
Base speed: 3,30 GHz
Sockets: 1
Cores: 8
Logical processors: 16

Fig. 13. Logical processors usage in multithread software

V. CONCLUSION

In this paper, we propose a numerically efficient algorithm that can help many researchers to reach real-time requirement in their programs for vessel detection. The obtained results show that the parallelization of the code is needed, so the parallel signal processing on CPU was performed. Additionally, the proposed method, does not need any specialized hardware, only a general-purpose computer.

ACKNOWLEDGMENT

This research, as a part of P.148 Project, was funded by Vlatacom Institute. The APC was funded by Vlatacom Institute. The research was also supported by the Serbian Ministry of Education, Science and Technological Development.

REFERENCES

[1] A.M. Ponsford and J. Wang, “A review of high frequency surface wave radar for detection and tracking of ships,” Turk J Elec Eng&Comp Sci, vol. 18, pp. 409–428, 2010.
[2] M. Jankiraman, “FMCW Radar Design,” Kindle ed., Artech House, England, 2018.
[3] K. Gurgel and T. Schlick, “Remarks on Signal Processing in HF Radars Using FMCW Modulation,” Proceedings of the International Radar Symposium IRS 2009, Hamburg, Germany, 2009.
[4] B. Kim, Y. Jin, J. Lee and S. Kim, “High-Efficiency Super-Resolution FMCW Radar Algorithm Based on FFT Estimation,” Sensors 2021, vol. 21, 4018. https://doi.org/10.3390/s21124018 , 2021.
[5] Liu, G.; Yang, W.; Li, P.; Qin, G.; Cai, J.; Wang, Y.; Wang, S.; Yue, N.; Huang, D. “MIMO Radar Parallel Simulation System Based on CPU/GPU Architecture,” Sensors 2022, vol. 22, 396. https://doi.org/10.3390/s22010396
[6] A.C. Sodan, J. Machina, A. Deshmeh, K. Macnaughton, “Parallelism via multithreaded and Multicore CPUs,” IEEE Computer Society, vol. 43, issue: 3, pp. 24-32, Mar. 2010.
[7] D. Dheeraj, B. Nitish, S. Ramesh, “Optimization of Automatic Conversion of Serial C to Parallel OpenMP,” International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover, PES Institute of Technology Bangalore, India, Dec. 2012.
[8] E. Ayguade, N. Copty, A. Duran, J. Hoeflinger, “The Design of OpenMP tasks,” IEEE Transactions on Parallel and Distributed systems, vol. 20, Issue: 3, pp. 404-418, June 2008.
[9] B. Kim, Y. Jin, J. Lee and S. Kim, “Low-Complexity MUSIC-Based Direction-of-Arrival Detection Algorithm for Frequency-Modulated Continuous-Wave Vital Radar”, Sensors 2020, vol. 20, 4295. https://doi.org/10.3390/s20154295
[10] M. Erić, B. Igrić, “Practical Implementation and Performance Estimation of MUSIC Method Implemented on Signal Processor TMS 320c30,” Scientific-Technical Review, vol. LIV, No.1, 2004.