U.P.B. Sci. Bull., Series C, Vol. 80, Iss. 2, 2018 ISSN 2286-3540

A LOW COMPLEXITY METHOD FOR ANALYZING ACOUSTIC ECHO SIGNALS IN THE FRAMEWORK OF A CELLULAR AUTOMATA VIRTUAL ENVIRONMENT

Mihai BUCURICA[1], Ioana DOGARU[2], Radu DOGARU[3]

This paper evaluates the accuracy of ultrasound-based object shape classification using a new low complexity method (Binary Transitions) in the context of our recent work on analyzing acoustic signals in virtual environments modeled with cellular automata, which is also reviewed in this paper. This low complexity feature extraction method is compared with other natural computing methods previously designed to optimize solutions for autonomous agents (e.g. robots) localization and orientation. The sound propagation virtual simulator (CANAVI), which was written in both Java and Python, is capable of emulating sound propagation in a controlled 2D environment using Cellular Automata. A review of the various CANAVI implementations and their computational performance is given in this paper. The classifier used in this paper is Fast Support Vector Classifier (FSVC), previously proved to be efficient in isolated speech recognition problems. The results in applying this low complexity transform, based on counting binary transitions, are encouraging in this particular context, showing very good accuracies at a low implementation complexity.

Keywords: ultrasound signal processing, CUDA, GPU, sound propagation simulator, Python, JIT, Cellular Automata, radial basis neural networks, complex nonlinear networks, support vector classifiers, ultrasound signal processing, kernel neural networks, object classification, reaction-diffusion transform, binary-transitions

1. Introduction

Object recognition is the ability to construct an artificial system performing as well as our own visual system, i.e. a system able to assign labels (e.g. nouns) to particular objects, ranging from precise labels ("identification") to course labels ("categorization"). Animals, such as bats or dolphins, use ultrasound signals to perceive the environment echoes to localize and orient themselves: they estimate

[1] Doctoral School on Electronics Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: mihai-ionut.bucurica@sdettib.pub.ro
[2] Natural Computing Lab. Dept. of Applied Electronics and Information Eng. University POLITEHNICA of Bucharest, Romania, e-mail: ioana.dogaru@upb.ro
[3] Natural Computing Lab. Dept. of Applied Electronics and Information Eng. University POLITEHNICA of Bucharest, Romania, e-mail: radu_d@ieee.org

the distance to the nearest object or even recognize the objects around them. Generally, the research in using ultrasound signals is mainly focused on range-to-obstacle determination. Our approach is different, in that we want to use the echoes as part of a large mechanism, aiming at tasks such as localization; previous studies [1] in the field of SLAM (Simultaneous Localization and Mapping) propose many algorithms [2] to discover and map the environment, but in general the most frequent methods include measurements from the sensors combined with the odometry information.

Furthermore, we demonstrate that echoes can be used for object recognition systems instead of tangible acoustic interfaces (TAIs) [3] [4]. TAIs represents innovative acoustic devices used for Human - Machine interaction which can classify or recognize objects that the human is in contact with. The applicability of those systems can range from devices for blind people to algorithms for robots to determine their position based on the tag-shaped objects they are in contact with.

A particularity of our approach is the construction of a cellular automata-based virtual environment modeling sound propagation in an editable scenario (with user defined obstacles, user defined sound sources and other freely adjustable parameters). Several versions of such simulators have been previously considered [5] [6] [7]. This approach allows to rapidly create databases associated to various tasks of interest (obstacle recognition, localization, etc.) resulting from applying specific feature extractors to echo signals [8][9].

The second particularity of our approach is the development and investigation of low complexity feature extractors given their specificity to be implemented in mobile autonomous agents with particular requirements for energy consumption. Herein we report novel results from employing for the first time a low complexity feature extractor BT [10] to the echo signals in order to detect object shapes. Also, results consider a newly introduced and efficient classifier, named FSVC [11] [12]. The resulting accuracies compare favorably to those obtained in our previous experiments [13].

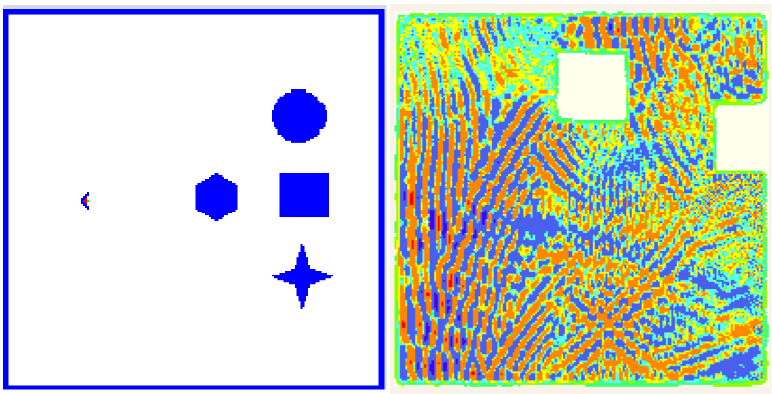
The virtual environment is presented in **Section 2**, the FSVC classifier is described in **Section 3**, a brief description of the Binary Transitions method is given in **Section 4**, along with the method for constructing feature vectors from echo signals. Classification results are summarized in **Section 5**.

2. Virtual environment architecture

The kernel of the virtual environment is based on the original idea described in [14] where the wave propagation equation is simulated using a simpler cellular automata model. An efficient sound propagation software for a 2D virtual scenario, named CANAVI, was written in C++ software language using tested theoretical equations [5] [6] for one dimensional wave. CANAVI was rewritten in both Java [13] and Python [15] in a successful attempt to optimize and add new functionality to the software.

The first version of the Java platform [7] includes a graphical user interface (GUI), part of which can be seen in Fig.1, and more capabilities such as artificial intelligence modules used to interpret signals from the environment as follows:

- A feature vector processing unit (FVPU) which transforms an arbitrary long echo (in our case the pressure recorded by the receiver) into a fixed size vector acting as input for the neural network;
- A database generator of feature vector collections and desired outputs specific to various kinds of experiments;
- The FSVC neural network, to be detailed in Section 4, capable of being trained using the database generator and capable of testing new echoes. Various recognition problems can be thus considered, depending on the particular choice of the database.



a) b)

Fig. 1 a): virtual scenario with obstacles (hexagon, star, circle and square) and the emitter/receiver probe; b) sound wave simulation run with the software (white spaces are the obstacles while the rest of the colors represents the amplitude of the sound)

Since CA are massively parallel, using hardware and programming tools supporting a high degree of fine-grained parallelism is beneficial in speeding-up computations, thus more diverse and complex virtual experiments can be performed on such a simulator. Consequently, in [15] we compared three implementations of the CA-based virtual environment:

a) The Java platform with some improvements. (version 2);
b) Python (using Anaconda distribution [16]) employing the Numba package, particularly using the JIT (Just In Time Compiler) for efficient use of the CPU.

c) Python with the NumbaPro (current versions require Numba instead) which allows us to use the GPU for computation acceleration.

As seen in Fig. 2, the improved Java implementation performs better than the Numba based Python. For large scenarios though, the NumbaPro Python version performs much better in terms of speed compared to other CA and CNN systems implemented using the same technology [17].

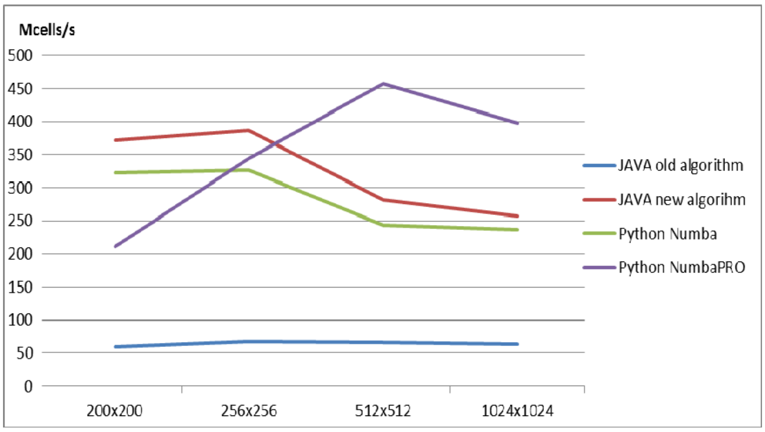


Fig. 2 Graphic comparing the results in terms of Mcells/s

3. Neural network architecture

The neural network used herein is the Fast Support Vector Classifier (FSVC) (or previous called RBF-M- modified Radial Basis Function), previously introduced in a series of papers [11][12][18][19]. It combines the guaranteed convergence learning of the output Adaline layer with a non-linear pre-processor made of several RBF units, thus providing the work of Adaline in an expanded space and consequently being able to represent complex data models. In other words, the network has the advantage of having simple and fast training (learning of the hidden layer reduces to a simple algorithm for selecting the *n* RBF centers among all samples in the training set) given a unique radius parameter. This radius is usually the only parameter to be optimized, although for a fine tuning the second parameter (threshold or overlapping factor *ov*) can be also considered.

The structure of the FSVC consists of two major aspects:

a) Fast initialization of the RBF centers chosen from the training set only in one training epoch;
b) The Gaussian RBF units are replaced with triangular ones as described in [11], without any penalty in terms of performance (modification made based on observation and confirmed by experiments: the shape of the RBF does not influence the overall accuracy of the NN).

A comparative study between FSVC and ELM was reported in [20]. We were interested to see if FSVC performs similarly to ELM, which is one of the most featured neural paradigms today. A wide range of problems was used leading to the following conclusions:

a) Accuracy is similar for both architectures in most cases;
b) The number of neurons appears to be smaller in most cases for FSVC;
c) ELM is slightly faster (in some cases ten times faster than FSVC) using the Matlab implementation, although is not entirely relevant since for FSVC there was no special effort to optimize the implementation.

To be noted that FSVC, in comparison to ELM, presents several advantages, particularly when targeting embedded systems (FPGA, microcontrollers, etc.):

a) All centers have a similar architecture because of the uniqueness of the radius, which makes it easy to multiply them;
b) There is no need for special resources (PRNG) to generate the centers since they are simply selected among the available data from the training set;
c) The pseudo-inverse learning algorithm in ELM (easy to implement in Matlab but less convenient for simpler hardware platforms) is replaced with the much simpler to implement iterative LMS algorithm.

In [21] we investigated if ELM (previously compared to FSVC), performs similarly to SVM. The main conclusions of the study [21] are:

a) Accuracy is similar for both architectures in most cases;
b) The number of neurons appears to be smaller in most cases for ELM;
c) For the same number of resources (neurons or SVs) the training time is higher for the ELM, but the testing time remains lower.

Concluding, our results in [20] and [21] confirm that FSVC is a reliable and fast neural network which can be easily implemented on hardware and consequently it was our final choice for embedding in the cellular automata based navigation virtual environment (CANAVI).

4. Binary Transitions method for sound preprocessing

4.1. Description

Binary Transitions is a method which proposes a low complexity algorithm, alternative to HMM- Hidden Markov Model. The method is best described in [10] and uses binary code to represent a signal. Here we use it for the first time in a different context (shape recognition), while it proved efficient for its low complexity in speech recognition task in [10]. It uses the idea of segmenting the sequence into a fixed number M of sub-sequences to conserve temporal features as described in [22].

If the echo sequence length is L, some M number of sub-sequences are chosen carefully (usually between 1 and 10), denoting the L/M sized sub-sequences as $\{S_k(t)\}_{t=1,L/M}$, where *k=1,..M* is the index of the sub-sequence. For each sub-sequence, *m* is a “bit-plane” index (e.g. $S_k^0(t)$ for *m=0* represents the most significant bit of $S_k(t)$**)** Consequently, such biplanes $b(t)=\{S_k^m(t)\}_{t=1,L/M}$ are next used using the algorithm in Section 4.2. to create parts of the feature vector. All vectors obtained for the *m* bit-planes and *M* sub-sequences are then concatenated, thus forming a unique feature vector which represents the particular sound echo to be next introduced in the learning process. A label for known echoes is also associated with each of the generated feature vectors.

4.2 Construct the feature vectors from echo signals

The inputs in this paper represented by the echoes have variable length, but this method produces fixed size feature vectors using the algorithm from Fig. 3 below:

```
t = 2;
Last = b(1);
Count = 1;
WHILE (t < T)
    IF ( b(t) <> Last )
        Last = b(t);
        Index = 1 + log2(count);
        IF (Index > n-1)
            Index=n;
        END
        F(Index)++; // increment by one
        Count=1;
    ELSE
        Count++; // increment by one
    END
END
Scale F (the feature vector) within [-1,1]
```

Fig.3 BT pseudo code

The variables in the algorithm above represent:
- F : Feature Vectors
- n : a parameter (ranges from 3 to 10, may optimize recognition accuracy here we took n=8)
- T : the duration of the binary sequence

- b(t) : the “t” sample (bit) of the binary sequence

5. Shape recognition results

Review of our previous results for shape classification: Using the first Java version of CANAVI [13] we conducted multiple experiments where object shape classification techniques are applied. Promising results were obtained: 100% accuracy for a 2-class system (circle and square) and **63.3% accuracy for a 4-class system** (square, circle, star and hexagon).

In a recent paper [23], we successfully applied a low complexity and efficient solution in terms of recognition accuracy for object shape classification, called Reaction Diffusion Transform (**RDT**), in a successful attempt to increase the accuracy **for the 4-class system** up to **91.6%** in some cases.

Results on exploiting the Binary Transitions (BT):

The experiments we conducted in this paper follow the same strategy as in the previous works:

- Construct the 4-class database system (circle, star, hexagon and square);
- Transform the raw signal into a fixed size annotated vector using the BT algorithm;
- Use FSVC for training and testing;
- Compare the results.

5.1. Construct the echoes database

For a fair comparison, we used the same Java version of CANAVI, in the same conditions as the previous work.

The database consists of recorded echoes originated from the ultrasonic waves generated inside the virtual simulator. The principle of recording each echo sample is the following:

a) A virtual scenario with a defined size is created. Besides the four scenarios used in the other reported solutions: 256 x 256 virtual environment, 200 samples (v1), 256 x 256 virtual environment, 400 samples (v2), 256 x 256 virtual environment, 1000 samples (v3), 512 x 512 virtual environment, 200 samples (v4), in this paper we increased the number at six by adding another two scenarios:
   - 512 x 512 virtual environment, 500 samples (v5)
   - 1024 x 1024 virtual environment, 500 samples (v6)

b) The probe and the obstacle are created. The obstacle is placed in front of the probe (same oy), but at random different distances (ox). The obstacle shaped is randomly chosen from the four shapes used;

c) The probe starts to emit an ultrasonic sound wave;

d) For N iterations, depending on the virtual environment size, the pressure in each cell is then computed as the wave is spreading. For large scenarios (larger than 1024 x 1024) the N is 2400, while for smaller scenarios (512 x 512, 256 x 256) the N is 1200;

e) At the end of the N iterations, the dynamics pressure in the point of the receiver representing the echo is recorded as a vector (sample for the database) and the obstacle used becomes the label for the sample.

5.2. Transform the recorded echo into a fixed size annotated vector using the BT algorithm

The feature extractor technique used herein has two roles:

1. Transform the echo into a fixed size annotated vector as needed by the classifier like any recognition system;
2. Reduce the size of the echo therefore reduce the computational overload for the classifier system.

Using the BT method described in Chapter 4, each echo has been reduced to a fixed size vector. Optimal parameters as n= 6 and M=10 have been tuned up testing multiple values ranging n between 1 and 10 and M between 2 and 15 as can be seen in Chapter 5.4

5.3. Use FSVC for training and testing

For each scenario the samples were divided into an equal number for training and testing. The best results were obtained for the v3 scenario: 256 x 256 environment, 1000 samples (500 for training and 500 for testing) using the FSVC classifier with the following parameters: distance used: Euclidian, learning rate (1/16), kernel: Gaussian.

5.4. Results

Multiple experiments with different values for the BT's parameters algorithm have been made, part of which can be seen in the Table 1, concluding with the fact that n=6 and M =10 is the optimal solution for the given problem and database (256 x 256, 1000 samples, 4-class).

Table 1

Accuracy in 4-class problem for different values of the n and M (BT's parameters)

Dataset	Scenario	Samples	n	M	Best accuracy (1=100%)
V1	256 x 256	200	2	2	0.56
V1	256 x 256	200	5	2	0.81

V1	256 x 256	200	6	3	0.78
V1	256 x 256	200	6	10	0.76
V2	256 x 256	400	8	4	0.695
V2	256 x 256	400	10	2	0.735
V2	256 x 256	400	6	2	0.785
V2	256 x 256	400	6	10	0.755
V3	256 x 256	1000	6	15	0.83
V3	**256 x 256**	**1000**	**6**	**10**	**0.88**
V3	256 x 256	1000	7	10	0.866
V3	256 x 256	1000	5	6	0.836
V4	512 x 512	200	4	2	0.45
V4	512 x 512	200	6	10	0.7
V4	512 x 512	200	6	2	0.55
V4	512 x 512	200	10	15	0.635
V5	512 x 512	500	6	10	0.78
V5	512 x 512	500	5	10	0.755
V5	512 x 512	500	10	5	0.54
V6	1024 x 1024	500	4	2	0.695
V6	1024 x 1024	500	6	10	0.735

Table 2 and Fig. 4 contain the comparison accuracy between the proposed solution, BT, other low complexity solution presented in [23] and the initial method [13] used for shape classification for the presented problem. For the 512 x 512 and 1024 x 1024 scenario, there is no data available for the original method.

Table 2

Accuracy comparison between RDT, BT and original method

Dataset	Scenario	Samples	RDT [23]	BT (this study)	In [13]
V1	256 x 256	200	85%	81%	61.2%
V2	256 x 256	400	89.5%	78.5%	64%
V3	**256 x 256**	**1000**	**91.6%**	**88%**	**66%**
V4	512 x 512	200	74%	70%	NA
V5	512 x 512	500	77%	74%	NA
V6	1024 x 1024	500	75%	73%	NA

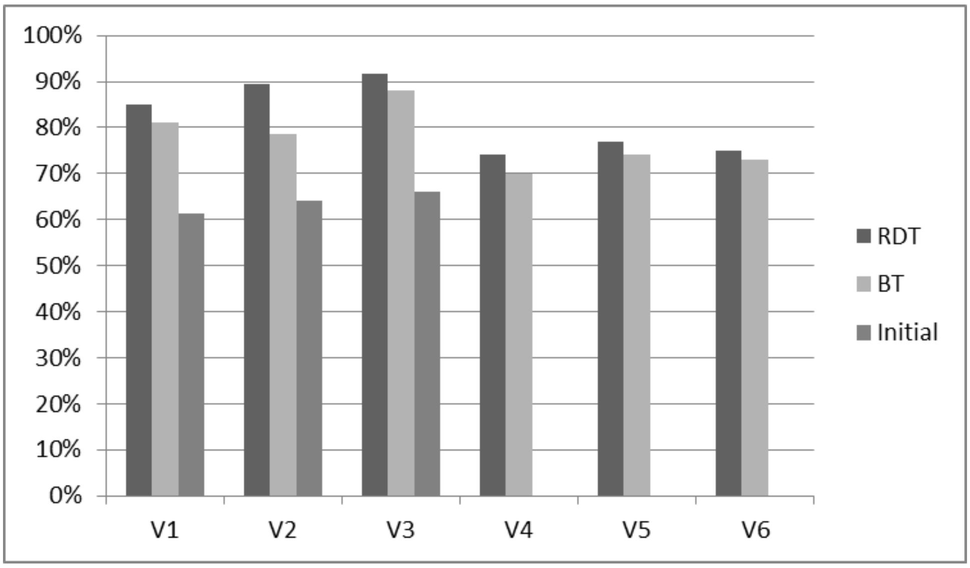


Fig.4 Accuracy results for RDT, BT and initial [13] method (v1, v2, v3, v4, v5, v6)

Table 3 contains comparison results regarding the computational complexity in terms of time of execution for each method presented. Execution time represents the time needed for the method to parse and transform an echo into a fixed size vector. Generating the echoes and using the FSVC classifier is irrelevant to the computational complexity because it is the same for all three methods. Only the feature extractor technique has different time. Execution time varies for the size of input vector (echo), so the comparison is made between the v5 and v6. Scenarios v1-v4 has the same vector size as v5. The values presented represent the average time.

Table 3

Computational complexity comparison between RDT, BT and initial [13] method

Dataset	Scenario	Iterations (Vector size)	RDT (ms)	BT(ms)	Initial (ms)
V5	512 x 512	1200	325	106	50
V6	1024 x 1024	2400	500	186	NA

6. Conclusions

For **the 2-class problem**, using square and circle, no optimization is required, all 3 algorithms: original [13], RDT [23] and BT (this study) reach maximum performance: **100%.**

For **the 4-class problem** the best accuracy obtained by the BT algorithm is **88%**, which is slightly lower than RDT [23], but better than the original [13]. Note that by its nature (counting bits), BT requires fewer arithmetic operations than RDT and will lead to much simpler hardware implementations while the loss in performance comparing to RDT is minimal.

In conclusion, this paper evaluates BT, a low complexity method for processing echo signals, showing that it achieves a good compromise between accuracy (only slightly worse than what is obtained with RDT, and much better than the initial feature extractor [13]) and computational complexity (better than for the RDT). This result indicates BT as a good choice for an embedded solution for recognition for an agent operating in a real environment (e.g. robot inside a building), where the agent can distinguish between a square shape obstacle and a circle one (for example), with 100% accuracy. Furthermore, the shape of that object can be interpreted as a tag and further used to determine the position of the agent in an indoor environment, if the position of the tag is also known. Our future research will consider in more detail such developments leading to an economical and accurate localization of agents based on ultrasound-echoes.

R E F E R E N C E S

[1] G. Tuna, K. Gulez, V. Gungor, T. Veli Mumcu, “Evaluations of different Simultaneous Localisation and Mapping(SLAM) algorithms”, published in IECON 2012- 38th Annual Conference on IEEE Industrial Electronics Society, October 2012 (pp. 2693-2698)

[2] Wolf, Denis F and Gaurav S. Sukhatme, “Mobile robot simultaneous localization and mapping in dynamic environments”, in Autonomous Robots, in Kluwer Academic Publishers, July 2005, (Volume 19, pp. 53-65)

[3] F. Antonacci, G. Prandi, G. Bernasconi, R. Galli, A. Sarti, “Audio-based object recognition system for tangible acoustic interfaces”, in Haptic Audio visual Environments and Games, (2009, November), IEEE International Workshop on (pp. 123-128).

[4] M. Bordegoni, F. Ferrise, S. Shelley, M. Alonso, D. Hermes, “Sound and tangible interface for shape evaluation and modification”, in Haptic Audio visual Environments and Games, October 2008, IEEE International Workshop (pages 148-153)

[5] R. Dogaru, I. Dogaru, "An Efficient Sound Propagation Software Simulator Based on Cellular Automata", in Proceedings ISEEE 2010 (September 2010, Galati, Romania), pp. 273-276.

[6] R. Dogaru, I. Dogaru, "Efficient realizations of a Sound Propagation Processor as a Cellular Nonlinear Network”, in Proceedings of 10th International Symposium on Signals, Circuits and Systems(ISCCS), 2011, Iasi, Romania, pp 165-168

[7] M. Bucurica, R. Dogaru, “Virtual environment using cellular automata for experimental studies of sound propagation and echo classification with artificial neural networks”, published in Fundamentals of Electrical Engineering (ISFEE) by the IEEE Computer Society, Nov. 2014 (pp. 1-4)

[8] Mahdi Tabassian, Nicola Testoni, Luca De Marchi, Francesca Galluzzo, Nicolo Speciale, Guido Masetti, "Learning Features from Medical Radiofrequency Ultrasonic Signals by Independent Component Analysis" in 2014 IEEE 27th International Symposium on Computer-Based Medical Systems, pp. 62 - 65

[9] Hsieh-Wei Lee, King-Chu Hung, Bin-Da Liu, Sheau-Fang Lei, Hsin-Wen Ting, "Realization of High Octave Decomposition for Breast Cancer Feature Extraction on Ultrasound Images", published in Circuits and Systems Part I: Regular Papers, IEEE, pp. 1287 - 1299

[10] R. Dogaru, "A Low Complexity Algorithm for Isolated Sound Recognition using Neural Networks", published in Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on. IEEE, 2007. p. 1-4.
[11] R. Dogaru, A.T. Murgan, S. Ortmann, M. Glesner, "A modified RBF neural network for efficient current-mode VLSI implementation", in Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (Micro-Neuro 96), IEEE Computer-Press, Lausanne 12-14 Feb. 1996, pp. 265-270
[12] R. Dogaru, I. Dogaru, "A super fast vector support classifier using novelty detection and no synaptic tuning," 2016 International Conference on Communications (COMM), Bucharest, 2016, pp. 373-376.
[13] M. Bucurica, R. Dogaru, "Shape object classification using echoes inside a virtual environment application", in Proceedings of ECAI2015, the 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, 2015. p. Y-13-Y-16.
[14] T. Komatsuzaki and Y. Iwata, "Study on Acoustic Field with Fractal Boundary using Cellular Automata", Proc. of ACRI2008 Springer LNCS5191, pp. 282-290.
[15] M. Bucurica, I. Dogaru, R. Dogaru, "Improving computational efficiency for implementing a sound propagation simulation environment using Python and GPU" in Electronics, Computers and Artificial Intelligence (ECAI), 2016 8th International Conference on (pp. 1-4).
[16] https://anaconda.org/anaconda/python
[17] R. Dogaru, I. Dogaru, "A Low Cost High Performance Computing Platform for Cellular Nonlinear Networks using Python for CUDA", published in 2015 20th International Conference on Control Systems and Computer Science, 27-29 May 2015, pp. 593 – 598
[18] R. Dogaru, "A hardware oriented classifier with simple constructive training based on support vectors", in Proceedings of CSCS-16, the 16th International Conference on Control Systems and Computer Science, May 22 - 26, 2007, Bucharest, Romania, Vol.1, pp. 415-418, ISBN 978- 973-718-741-3.
[19] R. Dogaru, I. Dogaru, "An Efficient Finite Precision RBF-M Neural Network Architecture Using Support Vectors", in Proceedings 10's Symposium on Neural Networks Applications in Electrical Engineering, NEUREL2010, Sept. 23-25, 2010, pp. 127-130.
[20] M. Bucurica, R. Dogaru, "A comparison between Extreme Learning Machine and Fast Support Vector Classifier," 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, 2015, pp. Y-9-Y-12.
[21] M. Bucurica, R. Dogaru, I. Dogaru, "A comparison of Extreme Learning Machine and Support Vector Machine classifiers", Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on. IEEE, 2015. pp. 471-474.
[22] R. Dogaru, G.N. Costache, 0. Dumitru, I. Gavat, "A nested temporal averaging feature extraction method for isolated word recognition using SVM classifiers" 2006, Galati, Romania, pp. 78-81.
[23] M. Bucurica, I. Dogaru, R. Dogaru, "A low complexity method based on reaction-diffusion transform for ultrasound echo-based shape object classification", *2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE)*, Galati, Romania, 2017, pp. 1-5.