BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI

Volumul 62 (66)

Numărul 3

AUTOMATICĂ şi CALCULATOARE

2016 **Editura POLITEHNIUM**

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
PUBLISHED BY
"GHEORGHE ASACHI" TECHNICAL UNIVERSITY OF IAŞI
Editorial Office: Bd. D. Mangeron 63, 700050, Iaşi, ROMANIA
Tel. 40-232-278683; Fax: 40-232-237666; e-mail: polytech@mail.tuiasi.ro

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volumul 62 (66), Numărul 3 2016

AUTOMATICĂ şi CALCULATOARE

S U M A R

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volume 62 (66), Number 3 2016

AUTOMATIC CONTROL and COMPUTER ENGINEERING

CONTENTS

.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 3, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

DEEP LEARNING APPLIED IN DISTRIBUTED IMAGE PROCESSING ARCHITECTURE

BY

MIHAI SCUTARU[*], BOGDAN BUDESCU, CODRUŢA ENE, FLORIN MOLDOVEANU and SORIN MORARU

Transilvania University Braşov,
Automatics and Information Technology Faculty

Received: June 9, 2016
Accepted for publication: July 14, 2016

Abstract. Recent advancements have proven that image processing and machine learning approaches are suitable for a large number of applications, assisting the end-user in an increasing number of critical applications. These range from intra-operative medical guidance to self-driving cars. Furthermore as data acquisition and large scale storage becomes more and more easily accessible, the degree to which automated algorithms can be trained to overcome specific problems has greatly increased. This vast amount of information has led to a new class of challenges for the scientist developing the algorithms. Traditionally each algorithm is engineered to extract a particular set of features from the input data and produce a model or present a decision to the user. Given the large variability in the input data, another approach is needed that doesn’t require a-priori analysis of the data. Recently the method which has proven to be more effective at automatic feature extraction is Deep Learning. This mechanism essentially employs multi-layer convolutional neural networks in order to provide automatic feature extraction from a large enough dataset. This approach has proven to become more effective as the amount of input data increases. In order to be able to efficiently employ such algorithms classic computing architectures have proven to be ineffective. This manuscript presents a novel, fully functional approach in which Deep Learning algorithms can be efficiently

[*] Corresponding author; *e-mail*: scutaru.mihai@gmail.com

trained using a distributed computing infrastructure. Starting from data storage, compute node execution, task arbitration and finally the presentation of the results, all the elements come together to provide the necessary environment for algorithm development.

Keywords: Machine-Learning; Deep Learning; Cloud computing.

2010 Mathematics Subject Classification: 68M99, 68W35.

1. Introduction

Machine-Learning is currently one of the most important and active research fields. It is essentially a sub-branch of computer science which has evolved from basic pattern recognition and is commonly known as a field of study that gives computers the ability to learn without being explicitly programmed. Lately, due to significant advances in computer architectures and widespread access to large computing power, the importance of machine-learning has become essential. More and more application have become driven by pure artificial intelligence, ranging from searching algorithms, speech processing in smartphones, up to computer aided surgery and self -driving cars. The disruptive impact of using these techniques can be observed in a large array of scientific fields and the more applications are developed using it, the understanding of its benefits become greater. Also including the concept of processing vast amounts of data (otherwise known as big-data applications) is quickly becoming the tool of choice of extracting relevant information from collections that have previously been too large to process otherwise.

Currently at the forefront of the machine-learning, a novel technique has emerged, in the form of Deep Learning (DL). Deep learning essentially refers to having a set of feature learning models which consist of multiple layers. Features are in essence different levels of abstraction of the raw input data. Different layers of a deep network learn different features by using many key neural network techniques and algorithms. Also deep networks exploit the fact that higher-level abstractions are significantly better at representing the input data than manually tailored features, thus achieving better overall performance (Wang *et al*., 2015).

The popular deep learning models can be grouped into three categories based on the connection types between layers (Wang *et al*., 2015).

1. Category A consists of feed-forward models wherein the layers are directly connected. The extracted features at higher layers are fed into prediction or classification tasks, *e.g.*, image classification (Krizhevsky *et al*., 2012). Example models in this category include Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN) and Auto- Encoders.

2. Category B contains models whose layer connections are undirected. These models are often used to pre-train other models (Hinton and

Salakhutdinov, 2006), *e.g.*, feed-forward models. Deep Belief Network (DBN), Deep Boltzmann Machine (DBM) and Restricted Boltzmann Machine (RBM) are examples of such models.

3. Category C comprises models that have recurrent connections. These models are called Recurrent Neutral Networks (RNN). They are widely used for modelling sequential data in which prediction of the next position is affected by previous positions. Language modeling (Mikolov *et al.*, 2011) is a popular application of RNN.

In this manuscript the main focus will be on Category A networks, moreover on the challenges faced when applying these techniques in large scale computing infrastructures. In Section 0, we present the challenges faced when training deep networks and propose a novel approach in distributing the work load efficiently in order to obtain improved results in a small time-frame. In Section 0 the results of our experiments are presented leading to the conclusions from Section 0.

2. The Deep Network Training Challenge

One important particularity of neural networks is the importance played by hyper parameters. These are a set of configuration parameters applied to the network which affect how a model is trained. The selection process of these parameters is important because the correct set will lead to high performance in a reduced amount of time, while bad choices will lead to prolonged training intervals and bad performance. Regularly, machine learning practitioners rerun the same model multiple times with different hyper parameter sets, in order to identify the best one. This technique is referred to as hyper parameter tuning. In the context of deep networks this process grows in complexity along with the size of the network. Each additional layer will need a set of hyper-parameters and this selection will affect both the individual layer, as well as the whole functionality of the network.

In general there are many parameters which need to be carefully selected. Two good examples are the number of neurons in each layer and the learning rate. In the case of the number of neurons, having too few neurons will reduce the expression power of the network, but too many will substantially increase the running time and induce noise in the estimated output. For the learning rate, if it is too high, the neural network will focus only on the last subset of samples seen, and ignore all experience accumulated before. If the learning rate is too low, it will take too long to reach a good state. There are various algorithms that can be applied to optimize this such as the ones describe in (Bergstra *et al.*, 2010; 2011), which are currently held as standard practice.

Another approach is presented in (Snoek *et al*., 2012) and takes a full Bayesian approach to the treatment of hyper parameters. This is done through Bayesian Optimization (Brochu *et al*., 2010) with Gaussian Process Priors. In this technique, the main interest is finding the minimum of a function $f(x)$ on a bounded set X which is a subset of R^D. The reason that Bayesian optimization is different from other procedures is that it constructs a probabilistic model for $f(x)$ and then exploits the model in to decide about where in X the next evaluation takes place, while integrating out uncertainty. The main objective of this approach is to use all the information available from previous evaluations of $f(x)$ and not rely on local gradient and Hessian approximations. The main result is a procedure that can find the minimum of a difficult non-convex function with what can be considered few evaluations, but with the cost of performing more computations to determine the position of the next iteration. In the cases in which the evaluations of $f(x)$ are expensive to perform, such as the case of training a multi-level neural network, it is easy to justify the extra computations needed to make better parameter decisions

Another very important aspect is the scalability of the training process itself. It has been shown that larger training datasets and bigger network models lead to better accuracy such as (Ciresan *et al*., 2010; Szegedy *et al*., 2015). But this tendency leads to massive training sets as well as processing requirements that oftentimes exceed the capacity of a single CPU-GPU set. Even if the hardware requirements fall within the scope of commodity server specifications, the time needed to process all inputs and provide results for a single training configuration may exceed any real world feasible limits. For instance it takes 10 days (Yadan *et al*., 2013) to train the DCNN (Krizhevsky *et al*., 2012) with 1.2 million training images and 60 million parameters using one GPU. Also according to the authors, even with 2 GPUs on the same machine the training still took about 6 days.

Currently, as the state of the art, there are several Deep Learning libraries designed to abstract a significant part of the underlying concepts in such a way that they facilitate automation through succinct specifications of tasks like describing intricate network architectures and complex training processes, and, as such, help leveraging the distribution of tasks in the context of environments such as SINGA (Wang *et al*., 2015). Examples of libraries can be: Caffe (Jia *et al*., 2014), Theano (Zaharia *et al*., 2010), or more recently TensorFlow (Abadi, M. et al., 2015). Also these can be used in conjunction with high performance clustering and optimization solutions such as Spark (Zaharia *et al*., 2010) or HTCondor (Thain *et al*., 2005).

In our experiments we will show the advantages of using Caffe in conjunction with HTCondor, to optimize the training process of deep network, with an application on the well-known MNIST training/testing datasets for handwriting recognition (Fig. 1).

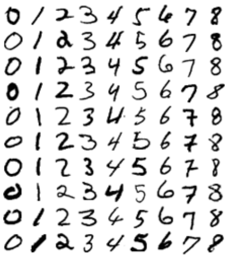


Fig. 1 – Sample of the MNIST dataset of handwritten digits.

3. Distributing the Learning Problem

For most untrivial problems that emerge in real life scenarios, solutions involving machine learning algorithms face the problem of searching the multidimensional meta-parameter space for a point with sufficiently low error. Traditionally, this time consuming, trial-and-error task based on educated guesswork is done manually, and requires a significant amount of effort from the expert performing the experiments. The process is prone to missing hard to guess co-dependencies in multidimensional space. Automated experiment planning tools like spearmint eliminate the need for an experienced (and lucky) human expert to spend time analyzing previous experiment results in order to decide at which point in meta-parameter space to evaluate the error next. However, the number of experiments that must be performed is still large in order to find a satisfactory model, especially when many meta-parameters are involved. Fortunately, the meta-optimization packages are able to leverage multiple computation nodes in order to parallelize the search in n-dimensional space.

Our solution builds the infrastructure for running multiple experiments in parallel in a scalable and efficient setting. We tested our approach using a naïve (exhaustive) search strategy on a heterogeneous cluster of six computers with various CPU models of the Intel Haswell family, amounts of system memory (4-8GB), GPU types (Nvidia Kepler and Maxwell) and amounts of GPU memory (2-4 GB). We used the HTCondor framework to schedule and manage the job execution. The scheduling policy has been to only assign a single job to each machine at a given time, as, due to the size of the networks, the GPUs of the machines were saturated during network training. Although the models and datasets we used were rather small, compared to the neural networks used to model recent problems of interest, the overhead for communication and task distribution was insignificant compared to the actual training time (*i.e*. seconds vs. minutes, for a single model). Also, at this stage we used local copies of the dataset. In the future, we will also experiment with distributed access to a single, shared, high-throughput datastore.

4. Experiments and Results

In order to assess the qualitative, domain specific advantages (*e.g.* in terms of classification accuracy) of being able to thoroughly explore the multidimensional search space of a family of convolutional network models induced by its set of meta-parameters, we designed a battery of simple experiments. We only chose a small set of key meta-parameters such as start learning rate, weight decay and various layers' sizes, either in terms of number of convolution kernels, or fully connected units. The latter two of these are able to be considered equivalent if a fully connected layer is interpreted as a convolutional layer, the kernel size of which happens to have the same shape as the input layer, as the search space increases exponentially in the number of meta-parameters.

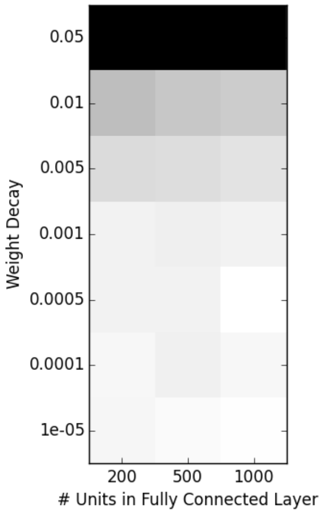


Fig. 2 – 2D slice in 5D error surface showing the effect of the interaction between the network's number of weights and regularization coefficient.

A cross-product of parameters' value vectors is preferred to a sequential independent search across each dimension, as the latter approach would hide the folds of the error surface caused by intertwined dependencies between meta-parameters. The most significant example of such interdependency can be usually observed on neural net models that employ regularization techniques, *e.g.* weight decay: nets with more need more regularization than ones with fewer parameters, but are, also, able to achieve better results; thus, searching for a good regularization coefficient for small nets would hide the significantly better performance achievable on larger ones, with a weight-decay appropriate for their size as shown in Fig. 2. The figure shows the accuracy achieved using combinations of meta-parameter values that influence the size and regularization of a simple LeNet-based model, when all other meta-parameters are fixed (initial learning rate =*1e-3,* 20 filters in the first layer, 100 filters in the second). Despite the noise caused by the intrinsic stochasticity of the learning process, induced by the random initialization of network weights before the start of each training session, and the jaggedness caused by slight decorrelation between training and test set, a clear maximum can be observed for the network using 1000 units in the hidden layer and a regularization coefficient of *5e-4*. Also, as stated above, the optimal weight decay coefficient increases proportionally with the network size. Another objective of the experiments is to measure and visualize the hypotheses of certain meta-parameter interactions, which are assumed by meta-parameter optimization packages, such as Spearmint (Snoek *et al*., 2012), which uses Gaussian processes to model the error as a function of the model's meta-parameters, thus implying the surface in question locally exhibits n-dimensional smoothness. In Fig. 2, the error map exhibits a set of gradients concentrically pointing to a theoretical maximum achievable with a network of infinite size and proportional regularization, thus validating the Gaussian surface model used by Spearmint.

For the purpose of generality and in order to keep the computational requirements at a feasible level even for a large enough number of trials, we chose to experiment with a family of models based on a simple, but pervasive model in the scientific community – Yann LeCun's LeNet convolutional architecture for classifying the handwritten digit images in the MNIST dataset. We chose to train and evaluate our models on the standard split of the dataset, as the purpose of our experiments is to illustrate the importance of correctly choosing meta-parameters of the model and to investigate the effects of their interactions on the results, rather than to attain a highly performant single model. Therefore, we used the test set only for cross-validation purposes, rather than for reporting a single model's generalization capacity.

The specifics of the search space explored in our experiments are defined by the sets of values for the following network/solver meta-parameters:

1. The initial learning rate of the stochastic batch gradient descent optimizer (which is further annealed by a power-law function during the optimization process), which ranges from *1e-1* to *1e-4* in exponential steps (of base 10);

2. The penalization coefficient for the L_2 norm of the network's weights, ranging from *5e-2* to *1e-5*, with an exponential stepping of base 2;

3. The number of distinct features learned by each layer, *i.e.* convolution kernels, for the first two convolution layers and number of units in the fully connected hidden layer, from the following sets, respectively, {10, 20, 50, 100}, {20, 50, 100, 200} and {200, 500, 1000}.

The test accuracy obtained by the base model, provided as example by the Caffe deep learning package is 24% larger than the best model found after thoroughly exploring the meta-parameter space (0.72% vs. 0.89%). As expected, the network that outperformed the original model has an overall larger number of parameters, but also shifts the peak of representation compression towards the deeper, more abstract layers. This is done by allowing for a highly overcomplete coding of local features in input images by using a larger number of convolution kernels in the first layer relative to the second one, therefore enabling the network to learn more succinct, dense and rich distributed representations more suitable for the final classification task.

5. Conclusions

The opportunity for machine-learning research scientists to better leverage high efficiency and high throughput computing resources allows them to push for qualitative leaps in the models they are able to synthesize in a given timespan, thus widening the horizons of their work to previously unfeasible applications. As a future development of the platform, we are experimenting with adapting automatic hyper-parameter tuning packages to this particular distributed computation infrastructure, in order to further decrease the time-to-market for machine-learning based products. This distributed environment also allows for further sharing the computational resources within scientific communities.

REFERENCES

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org. (2015).

Bergstra J., Breuleux O., Bastien F., Lamblin P., Pascanu R., Desjardins G., Turian J., Warde-Farley D., Bengio Y., *Theano: A CPU and GPU Math Compiler in Python*, In Proc. 9th Python in Science Conf., June, 2010, pp. 1-7.

Bergstra J.S., Bardenet R., Bengio Y., Kégl B., *Algorithms for Hyper-Parameter Optimization*, In Advances in Neural Information Processing Systems, pp. 2546-2554, 2011.

Brochu E., Cora V.M., de Freitas N., *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*, arXiv preprint arXiv:1012.2599 (2010).

Ciresan D.C., Meier U., Gambardella L.M., Schmidhuber J., *Deep, Big, Simple Neural Nets for Handwritten Digit Recognition*, Neural Computation, 22(12), pp. 3207-3220, 2010.

Hinton G.E., Salakhutdinov R.R., *Reducing the Dimensionality of Data with Neural Networks*, Science, 313(5786), pp. 504-507, 2006.

Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T., *Caffe: Convolutional Architecture for Fast Feature Embedding*, In Proceedings of the ACM International Conference on Multimedia, November, 2014, pp. 675-678.

Krizhevsky A., Sutskever I., Hinton G.E., *Imagenet Classification with Deep Convolutional Neural Networks*, In Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

Mikolov T., Kombrink S., Burget L., Černocký J.H., Khudanpur S., *Extensions of Recurrent Neural Network Language Model*. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, May, 2011, pp. 5528-5531.

Snoek J., Larochelle H., Adams R.P., *Practical Bayesian Optimization of Machine Learning Algorithms*, In Advances in Neural Information Processing Systems, pp. 2951-2959, 2012.

Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A., *Going Deeper with Convolutions*, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

Thain D., Tannenbaum T., Livny M., *Distributed Computing in Practice: The Condor Experience*, Concurrency and Computation: Practice and Experience, **17**, *2-4*, 323-356 (2005).

Wang W., Chen G., Dinh A.T.T., Gao J., Ooi B.C., Tan K.L., Wang S., *SINGA: Putting Deep Learning in the Hands of Multimedia Users*, In Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, October, 2015, 25-34.

Yadan O., Adams K., Taigman Y., Ranzato M.A., *Multi-GPU Training of Convnets*, arXiv preprint arXiv:1312.5853, **9**, 2013.

Zaharia M., Chowdhury M., Franklin M.J., Shenker S., Stoica I., *Spark: Cluster Computing with Working Sets*, HotCloud, **10**, p. 10, 2010.

ÎNVĂŢAREA PROFUNDĂ APLICATĂ ÎN ARHITECTURI DISTRIBUITE DE PROCESARE A IMAGINILOR

(Rezumat)

Progresele recente au dovedit că procesarea imaginilor şi abordările de învățare automatizată (machine learning) sunt potrivite pentru un număr mare de aplicații, asistând utilizatorul într-un număr tot mai mare de aplicații critice. Aceste aplicații au în vedere printre altele asistența activităților medicale intra-operative sau a mașinilor autonome. Mai mult, deoarece achiziția de date şi stocarea pe scară largă a informației devin mult mai accesibile a crescut semnificativ și capacitatea de a antrena algoritmii automatizați pentru a satisface probleme specifice. Această cantitate mare de informații a condus la o nouă clasă de provocări privind dezvoltarea algoritmilor. În mod traditional fiecare algoritm este proiectat pentru a extrage un anumit set de caracteristici din datele de intrare și de a produce un model sau să ofere o decizie utilizatorului. Dată fiind variabilitatea mare a datelor de intrare, este necesară o altă abordare care să nu impună o analiză a-priori a datelor. Recent, învățarea profundă (Deep Learning) s-a dovedit a fi o alternativă eficientă pentru extragerea automată a trăsăturilor. Acest mecanism utilizează în esență rețele neuronale convoluționale multi-strat, în scopul de a asigura extragerea automată a caracteristicilor dintr-un set de date suficient de mare.

Această metodă s-a dovedit a fi cu atât mai eficientă cu cât crește volumul datelor de intrare. Pentru a putea utiliza eficient astfel de algoritmi, se impune modificarea arhitecturilor de calcul deoarece cele clasice au devenit ineficiente pentru rularea noilor algoritmi. Această lucrare prezintă o abordare nouă, complet funcțională prin care algoritmii Deep Learning pot fi antrenați eficient utilizând o infrastructură de calcul distribuită. Pornind de la datele stocate, arbitrarea task-urilor, execuția la nivel de nod și în final prezentarea rezultatelor, toate aceste elemente formează un mediu necesar rulării algoritmilor.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 3, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

FIREWORKS ALGORITHM BASED SINGLE AND MULTI-OBJECTIVE OPTIMIZATION

BY

SILVIU-IOAN BEJINARIU[1,*], HARITON COSTIN[1,2], FLORIN ROTARU[1], RAMONA LUCA[1] and CRISTINA NIŢĂ[1]

[1]Institute of Computer Science, Romanian Academy Iaşi Branch,
[2]“Grigore T. Popa” University of Medicine and Pharmacy, Iaşi,
Faculty of Medical Bioengineering

Received: June 14, 2016
Accepted for publication: July 27, 2016

Abstract. Optimization procedures are often encountered in computer applications. These procedures can be more or less complex depending on the complexity and number of functions to optimize and also on the size of the problem domain and required precision. To obtain faster a near optimal solution the nature inspired algorithms were defined and used in the last years. In this paper it is studied the behaviour of Fireworks algorithm for both single and multi-objective optimization problems. The proposed strategy increases the number of Pareto optimal solutions generated given that a reduced number of individuals are used in this class of algorithms. The experiments were made using simple mathematical functions because the paper is focused on obtaining more precise solutions in a reduced processing time even if our final goal is to use the algorithm in image processing applications that require optimization procedures to be applied.

Keywords: Fireworks algorithm; optimization; Pareto solutions.

2010 Mathematics Subject Classification: 68T01, 68T20.

[*]Corresponding author; *e-mail*: silviu.bejinariu@iit.academiaromana-is.ro

1. Introduction

The nature-inspired optimization algorithms were developed in the last years with good performances in complex problems solving. Most of these algorithms are inspired from the survival, feeding or species perpetuation strategies of live beings: Bacterial Foraging (BFO), Bat (BA), Cuckoo Search (CSA), Firefly (FFA), Particle swarming (PSO) (Yang, 2014) or from other phenomena encountered in the daily life. Fireworks algorithm (FWA) which is inspired by observing fireworks explosion (Tan and Zhu, 2010) is such an example. These algorithms offer the possibility to approximate the optimal solution in lower computing time than other algorithms. Developed initially for single-objective optimization, almost all nature inspired algorithms were adapted for multi-objective optimization.

Our goal is to use these algorithms in image processing applications. Our previous approaches on BFO, BA and CSA based image registration and segmentation are presented in (Bejinariu *et al*., 2014; 2015a; 2015b). Other nature-inspired multi-objective optimization approaches are presented in: (Balasubbareddy *et al*., 2015; Chandrasekaran and Simon, 2012) for CSA (Niu *et al*., 2013; Sathya and Kayalvizhi, 2011; Guzman *et al*., 2010) for BFO, (Saeedi and Faez, 2011; Hu and Eberhart, 2002) for PSO (Liu *et al*., 2015) for FWA. Some of these are applied in image segmentation, as (Sathya and Kayalvizhi, 2011) and image fusion of multispectral images (Saeedi and Faez, 2011).

The paper is organized as follows. In the second section the Fireworks algorithm is described. The third section contains a short description of the multi-objective optimization problem and then it is proposed a strategy which allows using FWA in such optimization problems, by increasing the number of Pareto optimal solutions generated given that a reduced number of individuals are used in this algorithm. The fourth section describes the experiments on FWA usage for single and multi-objective minimization of some mathematical functions. The experiments were made using simple mathematical functions because the paper is focused on obtaining more precise solutions in a reduced processing time. The last section concludes the paper.

2. Fireworks Algorithm

The Fireworks Algorithm (FWA) was proposed in (Tan and Zhu, 2010; Tan, 2015) and it belongs to the Swarm Intelligence algorithms category. FWA is based on the fireworks explosion simulation. Like in all the other swarming algorithms, where the possible solutions are encoded as position of individuals that evolve in the problem domain, in this case the solutions positions are represented by the fireworks. The evolution is simulated by selection of sparks resulted in the fireworks explosion process. If the optimum value is unknown, the function to optimize is evaluated in all the reached positions during a number of steps and the position in which the best (minimum or maximum) position is obtained is considered as being the problem solution. Because the resulted sparks are randomly placed in the search space, this process is similar to a local search around each solution.

2.1. Algorithm Description for Single Objective Optimization

Single-objective optimization is described as the problem of determining the values of a parameter which minimize (or maximize) a function:

$$\min_{x \in S} f(x) \tag{1}$$

where S is a set of constraints.

In contrast to most swarming algorithms, in which individuals move in the problem domain trying to maximize their energy, in each iteration of FWA a variable number of descendants (sparks) is generated. To keep constant the number of individuals during their evolution, a global evaluation is required for all descendants.

Let's consider $X = \{x_i, i = 1, ..., N\}$ a set of N fireworks, and $x_i = \left(x_i^1, ..., x_i^d\right)$ their positions in the problem domain, where d is the dimension of the problem.

The general structure of the algorithm is described below as in (Tan, 2015):

```
Generate $X = \{x_i, i = 1,...N\}$ a randomly placed set of fireworks
Compute fitness for each firework $x_i$
while the stop condition is not met
    for each firework $x_i \in X$
        Explosion strength: Compute $S_i$, the number of sparks for each firework
        Explosion amplitude: Compute $A_i$, the sparks amplitudes for each firework
    end for
    Displacement: Generate regular sparks
    Evaluate sparks and check for the best solution
    Mutation: Generate Gaussian sparks
    Evaluate sparks and check for the best solution
    Selection of the best, the worst and other $N - 2$ random sparks as new fireworks
end while
```

2.2. Strategies used in FWA

The strategies used in each step of FWA are important for the algorithm's performances. The first section of the main loop in the algorithm refers to fireworks explosion modeling, *i.e.* generation of the next fireworks generation. The main explosion characteristics which must be considered are strength, amplitude and displacement.

Explosion Strength: the number of sparks S_i for each firework is computed as:

$$S_i = m * \frac{f_{worst} - f(x_i) + \varepsilon}{\sum_1^n \left(f_{worst} - f(x_i) \right) + \varepsilon} \tag{2}$$

where m is the total number of sparks, f_{worst} is the worst fitness value of the current fireworks set, $f(x_i)$ is the fitness value of the i^{th} firework and ε is used to avoid zero-division operations. To keep the number of sparks in a reasonable interval, minimum and maximum values that depend on m (total number of sparks) are imposed. To reduce unnecessary computing, less sparks are generated as the firework is closer to the worst solution.

Explosion Amplitudes A_i of the sparks obtained after the i^{th} firework explosion is computed as:

$$A_i = \overline{A} * \frac{f(x_i) - f_{best} + \varepsilon}{\sum_1^n \left(f(x_i) - f_{worst} \right) + \varepsilon} \tag{3}$$

where $\overline{A}$ is the desired sum of amplitudes, f_{best} is the best fitness value in the current fireworks set, $f(x_i)$ is the fitness value of the i^{th} firework and ε is used to avoid zero-division operations. To refine the best solution and to avoid the algorithm convergence in local solutions, the amplitude is lower for fireworks closer to the best solution and higher for those placed away from it.

In the *Regular sparks generation* step, sparks are generated for each firework and their new positions have to be computed.

Displacement Operation computes the position of each generated spark. For each spark obtained from the i^{th} firework, some randomly chosen coordinates are modified by adding a uniform random number in the interval $\left(-A_i, A_i\right)$.

$$x_i^k = x_i^k + U \tag{4}$$

where k are the randomly chosen coordinates and U is the uniform random value.

Mutation: To assure the diversity of the generated sparks, a number of sparks are generated from randomly chosen fireworks. The new positions are computed as:

$$x_i^k = x_i^k * g \tag{5}$$

where *k* are the randomly chosen coordinates and *g* are random numbers with Gaussian distribution.

After displacement and/or mutation are applied the position of the new sparks can be outside the problem domain. These sparks have to be re-positioned inside the domain, either in new random positions or in position computed using specific strategies.

Evaluation: the fitness function is evaluated in the all new sparks, in order to find a solution of the problem and also to select the next fireworks generation.

Selection strategy. As it was specified before, the number of fireworks remains constant during the evolution. The best and the worst sparks are retained for the next generation, and the other $N-2$ fireworks are randomly selected from the entire sparks set. The worst spark is selected to assure the diversity of the population and premature convergence of the algorithm to local solutions.

3. Multi-Objective Optimization Using FWA

Multi-objective optimization problems are usually encountered in decision making. Because the number of possible solutions is large and it is possible that an absolute optimal solution does not exist, multi-objective optimization is used only to approximate the possible solution set from which the decision maker will choose one.

Multi-objective optimization is described as the problem of determining the values of a parameter which minimize (or maximize) more than one function.

$$\min_{x \in S} \left[f_1(x),\ f_2(x), \ldots f_n(x) \right], n > 1 \tag{6}$$

where *S* is a set of constraints. In most cases, the values of the objective functions $f_i(x)$ can’t be minimized in the same time.

A solution $x^* \in S$ is *Pareto optimal* for the optimization problem if

$$\begin{gathered} \forall\ x' \in S, x' \neq x^*, \exists\, i \in \{1 \ldots \mathrm{n}\}, 1 \le i \le n \ni\ f_i(x') > f_i(x^*) \text{ or} \\ f_i(x') > f_i(x^*)\ \forall\ \ i \in \{1 \ldots \mathrm{n}\} \end{gathered} \tag{7}$$

A solution $x^* \in S$ is *weak Pareto optimal* for the optimization problem if and only if

$$\nexists\ x' \in S \ni f_i(x') < \mathrm{f}_i(x^*)\ \forall\ i, 1 \le i \le n$$

A solution $x^* \in S$ is *strict Pareto optimal* for the optimization problem if and only if

$$\nexists\ x^{\uparrow'} \in S \ni \left[f_\downarrow i\,(x^{\uparrow'}) \le f_\downarrow i\,(x^{\uparrow*})\ \forall\ i, 0 < i < n\ \ AND\ \exists\, i \ni \left|([f])\right|_\downarrow i\,(x^{\uparrow'}) < f_\downarrow i\,(x^{\uparrow*}) \right] \tag{8}$$

The Pareto Front is the set of all the efficient solutions. All the points in the Pareto front are called non-dominated points.

3.1. Methods for Multi-Objective Optimization Problems Solving

The simplest method to solve the multi-objective optimization problem is the *scalarization*. The problem is transformed in a single-objective optimization problem in which the objective is computed as a weighted sum of the objective functions:

$$\min \sum_1^n \alpha_i f_i(x)\text{, where } \alpha_i > 0, \forall\ i \text{ and } \sum_1^n \alpha_1 = 1 \tag{9}$$

The solution obtained by solving this problem is on the Pareto front: weak Pareto optimal if at least one $\alpha_i = 0$ and strict Pareto optimal if $\alpha_i > 0, \forall\ i = 1, ..., n$.

Another method to solve the multi-objective optimization problem named $\varepsilon -$ *constraints* is to optimize a single objective function while all other are transformed in restrictions:

$$\min f_k(x), k \in \{1, ..., n\} \text{ and } f_\downarrow i(x) \le \varepsilon_\downarrow i,\ \forall i \in \{1, ..., n\} \setminus \{k\} \tag{10}$$

In contrast to the *scalarization* method, the $\varepsilon -$ *constraints* has the advantage that it can achieve efficient solution in a non-convex Pareto front.

Goal Programming is another technique used in multi-objective optimization which substitutes the multiple criteria by restrictions added for each objective. In the *Multi-level Programming* approach the objectives are sorted using a priority criterion and optimal solutions are sought for each criterion between the optimal solutions of the previous higher priority criterion.

3.2. Fireworks Algorithm for Multi-Objective Optimization

The Fireworks algorithm can be adapted for multi-objective optimization as all other nature inspired algorithms. The main problem is that the algorithm must not only to choose the best solution but also in the selection step to find besides the best spark also the worst spark. In addition to solutions mentioned above, the dynamic neighborhood method is also used in multi-objective optimization (Fan, 2013). In short it consists of finding the best solution in terms of one objective function and then between the solutions having the value of the same objective in a neighborhood of the best one, the solution with the best value of the second objective is chosen. This method has the disadvantage that it requires sorting operations which may be globally time-consuming.

We propose to modify the Fireworks algorithm by:

– replacing the best known solution by a list of Pareto optimal solutions, in which all non-dominated intermediary solutions are stored;

– in each iteration, after sparks generation, they are checked for domination and stored in separate lists,

– the selection of next fireworks generation:

– instead of best firework, a non-dominated firework is randomly chosen,

– instead of worst firework, a dominated firework is randomly chosen,

– all other fireworks in the next generation are chosen from the non-dominated solutions list,

– in the evaluation step, all non-dominated sparks are added in the list of Pareto solutions which is checked for dominance and the dominated temporary solutions are removed from list.

In standard implementations of other nature inspired multi-objective optimization algorithms, in which the number of individuals that evolve in the problem domain is high, the selected individuals in each iteration are added to the current Pareto solutions list if are non-dominated by other solutions. In FWA, where usually the number of fireworks is reduced, the proposed approach allows obtaining more Pareto optimal solutions. In the next paragraphs, some experiments on the FWA usage are presented.

4. Experiments

Our goal is to use FWA for single and multi-objective optimization in image processing problem. Because these problems have high complexity of cost functions to evaluate, in this first step we decided to check the algorithm’s behavior for some simple mathematical functions.

4.1. Single-Objective Optimization Using FWA

For FWA evaluation in case of single-objective optimization were chosen three functions with different number of local and global extrema values in the multidimensional space. The same functions were used in (Bejinariu *et al.*, 2015b) for Cuckoo Search and Bat Optimization algorithms evaluation.

The three functions are defined as:

$$f_i(x):[-1;+1]^n \to R,\ i=1,\ 2,\ 3 \quad f_1(X)=\sum_{i=1}^{n}(x_i)^2 \tag{11}$$

$$f_2(X)=\sum_{i=1}^{n}(\sin(5*x_i))^2 \tag{12}$$

and

$$f_2(X)=f_1(X)+f_2(X)=\sum_{i=1}^{n}\left((x_i)^2+(\sin(5*x_i))^2\right). \tag{13}$$

where $X=(x_1,\ldots,x_n)\in[-1;+1]^n$ and n is the optimization problem dimension. In Fig. 1, the graphs of the three functions in case $n=2$ are presented.

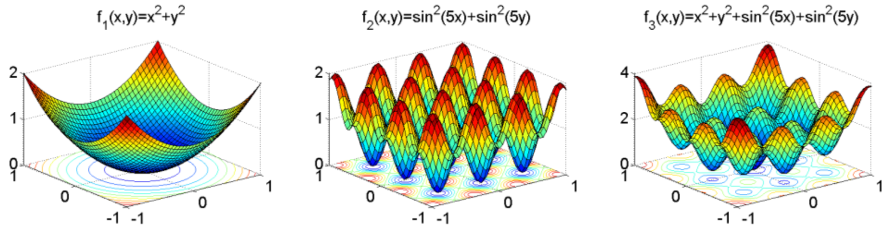


Fig. 1 – Graphs of the three function used for single-objective optimization test (Bejinariu *et al.*, 2015b).

In case $n=2$ all three functions reach their minimum value of 0 in $(0;\ 0)$. The second function reaches its minimum in nine different positions and only one of these will be determined by the optimization algorithm. In case of the third function which has also other eight local extrema points the algorithm must avoid the convergence to these values (Bejinariu *et al.*, 2015b). The parameters of the algorithm were chosen as follows:number of fireworks = 5, maximum amplitude = 40, regular sparks factor = 10 and number of Gaussian sparks = 5. The number of epochs in the evolution was varied in order to check its influence on the result precision: 250, 2500 and 25000. Table 1 shows the obtained results for each function and for two different dimensions of the problem domain: 5 and 15. The columns *Error* show the minimum values reached during the evolution, the *#Evals* columns show the number of cost function evaluations and *Best* columns shows the number of the evaluation in which the optimum value was reached. As mentioned above, it is possible that some local solutions may be obtained for the third function minimization. These cases are marked by (*).

A reduced number of 250 epochs allows obtaining good results in case of functions with a single extrema point. In case of multiple extrema the number of epochs must be increased. It must be noticed that for lower dimensions of the search space $n = 2, 3$, the correct solution is obtained also for reduced number of epochs. Also, in most cases, the best value was obtained in the latest iterations which indicate that the solution may be improved by increasing the number of epochs.

Comparing to the results presented in (Bejinariu *et al.*, 2015b), the CSA algorithm offers better results, but in case of FWA, about one-third of the cost function evaluations in CSA are enough to obtain reasonable values.

Table 1
Results Obtained in Case of Problem Dimensions n=5 and 15

	n	5			15		
Epochs	*f*	Error	#evals	Best	Error	#evals	Best
250	1	7.47E-08	14361	14090	7.40E-06	14340	14326
	2	1.91E-06	14337	14285	2.70E-04	14333	14240
	3	3.80E-01 *	14277	13791	3.04E+00 *	14330	14272
2500	1	3.67E-10	142967	142796	3.60E-07	143297	140500
	2	4.39E-08	142798	135432	2.36E-05	143483	137004
	3	4.73E-08	142793	134773	1.52E+00 *	143407	142992
25000	1	9.69E-15	1427596	1425498	5.72E-09	1439395	1435526
	2	4.49E-15	1429865	1355307	6.27E-07	1439570	1433576
	3	9.94E-15	1429038	1425702	1.01E-06	1439937	1424610

4.2. Multi-Objective Optimization Using FWA

Because the three functions used in the previous section reach the minimum value in the same position, for the evaluation of FWA based multi-objective optimization were chosen other two functions, defined as:

$$f_i(x) : [-1; +1]^n \to R,\ i = 4, 5$$

$$f_4(X) = \sum_{i=1}^{n} (x_i + 0.4)^2 , \qquad (14)$$

$$f_5(X)=\sum_{i=1}^{n}\left(\sin\left(4*x_i\right)\right)^2 \tag{15}$$

where $X=\left(x_1,...,x_n\right)\in\left[-1;+1\right]^n$ and n is the optimization problem dimension. In Fig. 2, the graphs of the three functions in case $n=2$ are presented.

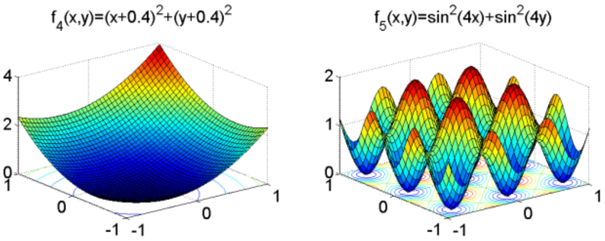


Fig. 2 – Graphs of the two function used for multi-objective optimization test.

4.2.1. Multi-objective optimization using the scalarization method

The scalarization method is the first one used for multi-objective optimization. The single-objective function used in optimization is: $\min\left(\alpha f_1(x)+(1-\alpha)f_2(x)\right)$ where $\alpha=0.1*k,\ k=\overline{1,10}$. In the figure below, the graphs of some functions computed for different values of α are presented.

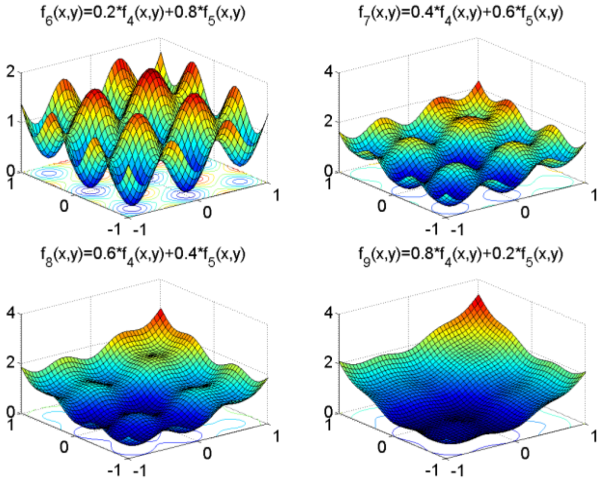


Fig. 3 – Graphs of the function to minimize computed for α = 0.2, 0.4, 0.6 and 0.8.

The parameters of the algorithm were chosen as follows: number of fireworks = 5, maximum amplitude = 40, regular sparks factor = 10 and number of Gaussian sparks = 5. The number of epochs in the evolution was varied in order to check its influence on the result precision: 2500, 25000 and 250000. Table 2 show the obtained results for the problem dimension $n = 5$.

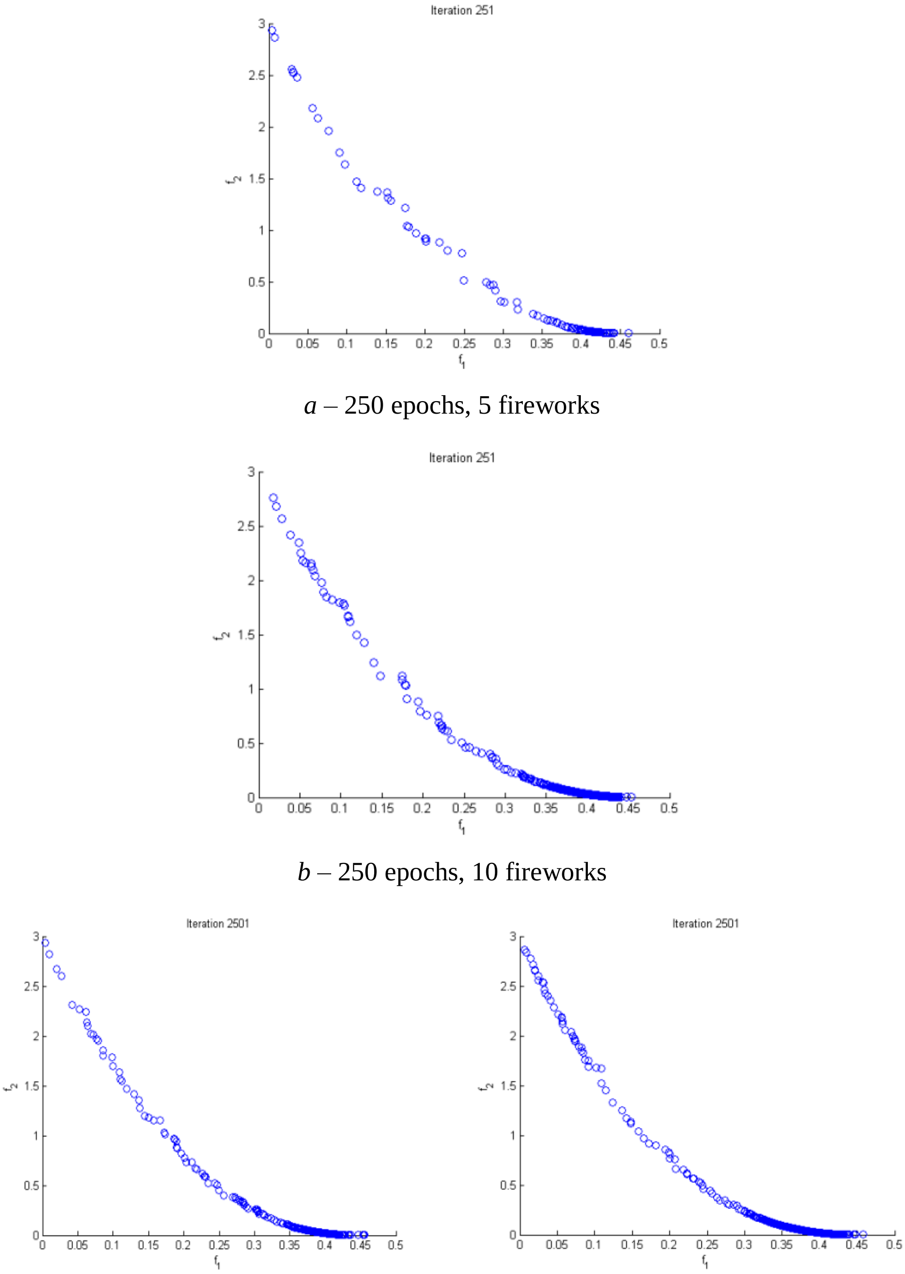
Table 2
Results Obtained in Case of Problem Dimension n=5

α	2500 epochs		25000 epochs		250000 epochs	
	fitness 1	fitness 2	fitness 1	fitness 2	fitness 1	fitness 2
0	7.03E+00	1.46E-11	7.03E+00	4.09E-14	7.03E+00	4.22E-22
0.1	7.78E-01	5.78E-04	7.55E-01	5.82E-04	7.44E-01	5.74E-04
0.2	7.54E-01	2.89E-03	7.42E-01	2.91E-03	7.31E-01	2.86E-03
0.3	7.48E-01	8.71E-03	7.26E-01	8.35E-03	7.15E-01	8.22E-03
0.4	7.16E-01	1.99E-02	7.05E-01	1.97E-02	6.84E-01	1.91E-02
0.5	6.88E-01	4.35E-02	6.78E-01	4.27E-02	6.68E-01	4.21E-02
0.6	6.48E-01	9.28E-02	6.39E-01	9.15E-02	6.20E-01	8.87E-02
0.7	5.70E-01	2.02E-01	5.62E-01	1.99E-01	5.62E-01	1.99E-01
0.8	4.75E-01	5.33E-01	4.68E-01	5.23E-01	4.61E-01	5.14E-01
0.9	2.38E-01	2.04E+00	2.35E-01	1.95E+00	2.35E-01	1.95E+00
1	8.41E-09	5.00E+00	2.62E-15	5.00E+00	4.23E-21	5.00E+00

It is obvious that for $\alpha = 0$ (the first row in Table 3), the function to minimize is f_2 which has eight solutions. For $\alpha = 1$ (the last row in Table 3), the function to minimize is f_1 which has an unique minimum in the interval $(-0.4; -0.4)$. A conclusion that can be drawn by analyzing the results presented in Table 3 is that the approximated values of the two functions are similar regardless the number of iterations, excepting the cases in which $\alpha = 0$ or $\alpha = 1$. So, the precision is not improved too much for exaggerate number of iterations. In this case, it seems that 2500 epochs are enough for acceptable results.

4.2.2. Multi-objective optimization using random selection of non-dominated solutions

In the following experiment, that uses the method proposed in the previous section, the varied parameters were: the number of epochs 250, 2500 and 25000 and the number of fireworks 5, 10 and 15. The other parameters were chosen as before: maximum amplitude = 40, regular sparks factor = 10 and number of Gaussian sparks = 5. In Figs. 4 and 5, the results obtained for the problem dimensions $n = 3$ and $n = 5$ are presented.



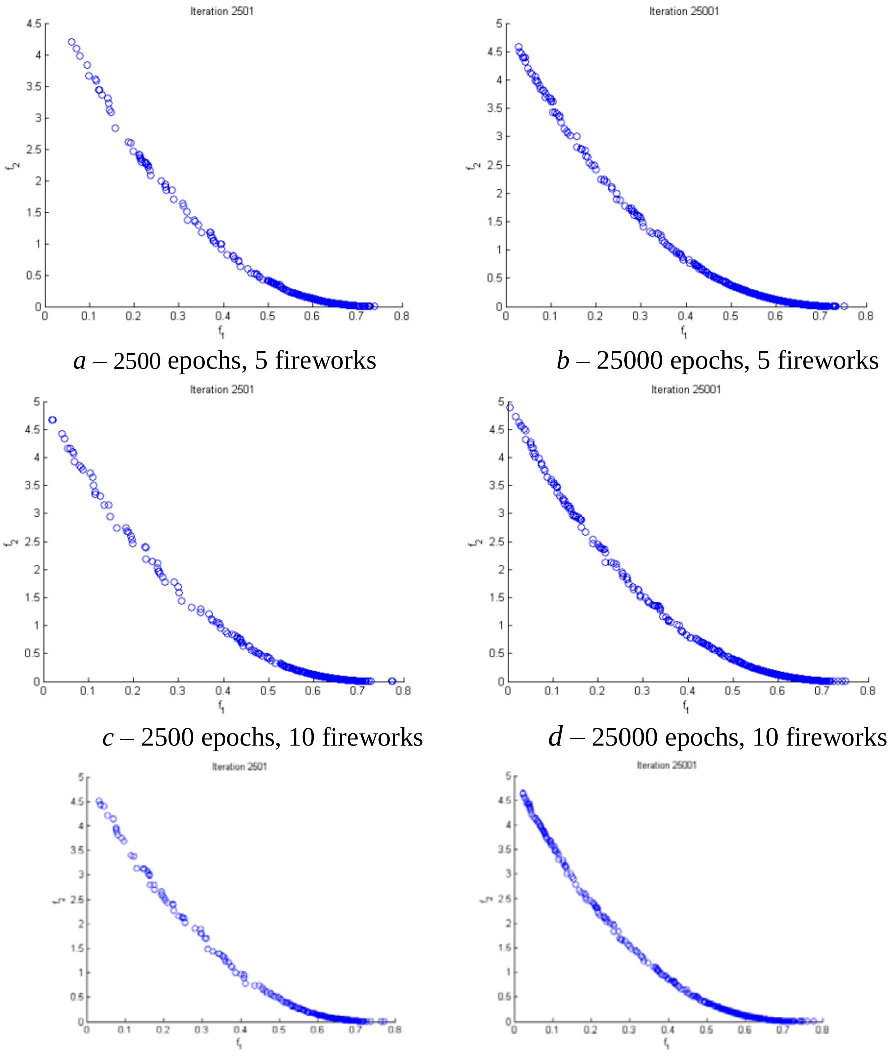
a – 250 epochs, 5 fireworks

b – 250 epochs, 10 fireworks

c – 2500 epochs, 5 fireworks

d – 2500 epochs, 10 fireworks

Fig. 4 – Pareto front obtained using FWA based multi-objective optimization in case *n=3*.



a – 2500 epochs, 5 fireworks

b – 25000 epochs, 5 fireworks

c – 2500 epochs, 10 fireworks

d – 25000 epochs, 10 fireworks

e – 2500 epochs, 15 fireworks

f – 25000 epochs, 15 fireworks

Fig. 5 – Pareto front obtained using FWA based multi-objective optimization in case *n*=*5*.

The number of non-dominated solutions in the Pareto front found during the fireworks algorithm is presented in Table 3.

Because the number of fireworks is reduced, the number of non-dominated solutions is also reduced. For the same reason, it may be noticed also that the functions values domain is not fully covered and the solutions are concentrated in some sub-intervals. Greater number of fireworks and iterations allows to obtain more non-dominated solutions at least for the higher dimensional problems. Concerning the number of cost function evaluations it can be seen that it increases fast with the number of iterations. In this experiment, the functions to minimize are enough simple and fast computed. In real optimization problems, like image registration where each evaluation requires to apply a geometric transform to the image and then to compute a similarity measure for each objective function (Bejinariu *et al.*, 2014; Bejinariu *et al.*, 2015a), the great number of evaluation can increase too much the time required by the optimization process. It's obvious that in these cases, only a reduced number of iterations will be acceptable.

Table 3

Number of Non-Dominated Solutions and Cost Functions Evaluations

n	Epochs	Fireworks	Solutions	# cost eval
3	250	5	247	14324
		10	770	27569
	2500	5	332	143167
		10	1614	275560
5	2500	5	272	143077
		10	441	275462
		15	987	412641
	25000	5	1493	1430859
		10	1986	2753613
		15	2361	4125732

5. Conclusions and Future Work

An analysis of the feasibility of the Fireworks algorithm in both single and multi-objective optimization problems is presented. In case of single objective optimization the results are less precise than other optimization algorithms, but the advantage of FWA is that a good result may be obtained in a reduced number of iterations leading to a more rapid problem solving. In case of the multi-objective optimization, a strategy for the fireworks selection is proposed. The experiments presented in the paper lead to the conclusion that for complex optimization problems it is better to choose reduced values for the parameters that influence the number of objectives evaluations.

The two versions of the Fireworks algorithm were implemented in C++ as Windows applications using the Microsoft Visual Studio 2015 framework. The obtained results were transferred to a Matlab application used to display the graphs presented in figures. The author's main interest is to use the nature inspired algorithms in image processing optimization applications as image registration or segmentation. The work presented in this paper is a first step of developing the algorithms that will be used in this kind of applications.

REFERENCES

Balasubbareddy M., Sivanagaraju S., Suresh C.V., *Multi-Objective Optimization in the Presence of Practical Constraints Using Non-Dominated Sorting Hybrid Cuckoo Search Algorithm*, Elsevier, Engineering Science and Technology, an International Journal, **18**, pp. 603-615, 2015.

Bejinariu S.I., Costin H., Rotaru F., Luca R., Niţă C., *Social Behavior in Bacterial Foraging Optimization Algorithm for Image Registration*, Proceedings of the 18th International Conference on System Theory, Control and Computing, Sinaia, Romania, October 17-19, 2014, pp. 330-334.

Bejinariu S.-I., Costin H., Rotaru F., Luca R., Nita C.D., *Automatic Multi-Threshold Image Segmentation Using Metaheuristic Algorithms*, Signals, Circuits and Systems (ISSCS), 2015 International Symposium on, Iași, July 9-10, 2015, pp. 1-4, 2015a.

Bejinariu S.-I., Costin H., Rotaru F., Luca R., Nita C.D., *Image Processing by Means of Some Bio-Inspired Optimization Algorithms*, Proc. of the IEEE 5th Int. Conference on E-Health and Bioengineering – "EHB 2015", Iași, Romania, pp. 1-4, 2015b.

Chandrasekaran K., Simon S.P., *Multi-Objective Scheduling Problem: Hybrid Approach Using Fuzzy Assisted Cuckoo Search Algorithm*, Elsevier, Swarm and Evolutionary Computation, **5**, pp. 1-16, 2012.

Fan K., You W., Li Y., *An Effective Modified Binary Particle Swarm Optimization (mBPSO) Algorithm for Multi-Objective Resource Allocation Problem (MORAP)*, Applied Mathematics and Computation, Elsevier, **221**, pp. 257-267, 2013.

Guzman M.A., Delgado A., De Carvalho J., *A Novel Multiobjective Optimization Algorithm Based on Bacterial Chemotaxis*, Engineering Applications of Artificial Intelligence, **23**, pp. 292-301, 2010.

Hu X., Eberhart R., *Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization*, Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, pp. 1677-1681, 2002.

Liu L., Zheng S., Tan Y, *S-Metric Based Multi-Objective Fireworks Algorithm*, 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1257-1264, 2015.

Niu B., Wang H., Wang J., Tan L., *Multi-Objective Bacterial Foraging Optimization*, Elsevier, Neurocomputing, 116, pp. 336–345, 2013.

Saeedi J., Faez K., *A New Pan-Sharpening Method Using Multiobjective Particle Swarm Optimization and the Shiftable Contourlet Transform*, Elsevier, ISPRS Journal of Photogrammetry and Remote Sensing, **66**, pp. 365–381 (2011).

Sathya P.D., Kayalvizhi R., *Optimal Multilevel Thresholding Using Bacterial Foraging Algorithm*, Elsevier, Expert Systems with Applications, **38**, pp. 15549-15564, 2011.

Tan Y., Zhu Y, *Fireworks Algorithm for Optimization*, in Tan Y., Shi Y., and Tan K.C. (Eds.), ICSI 2010, Part I, LNCS 6145, pp. 355-364, 2010.

Tan Y., *Fireworks Algorithm. A Novel Swarm Intelligence Optimization Method*, Springer-Verlag, 2015.

Yang X.-S., *Nature-Inspired Optimization Algorithms*, Elsevier Inc., 2014.

OPTIMIZARE UNI ŞI MULTI-OBIECTIV FOLOSIND ALGORITMUL FIREWORKS

(Rezumat)

Algoritmii de optimizare sunt utilizaţi în numeroase aplicaţii software. Adesea, identificarea soluţiilor optime este foarte dificilă datorită numărului mare de variabile dar şi complexităţii funcţiilor fitness. Algoritmii meta-euristici de inspiraţie naturală reprezintă o variantă pentru obţinerea unor soluţii apropiate de cele optime folosind resurse computaţionale mai reduse, deoarece simulează modelul eficient de hrănire, supravieţuire sau perpetuare a speciei, al fiinţelor vii sau alte modele ale unor fenomene din viaţa de zi cu zi.

Această lucrare este dedicată studiului comportamentului algoritmului Fireworks în probleme de optimizare uni şi multi-obiectiv. Scopul principal al autorilor este de a utiliza acest tip de algoritmi în aplicaţii de procesare a imaginilor, lucrarea de faţă fiind doar un prim pas în utilizarea modelului respectiv.

În ceea ce priveşte optimizarea uni-crierială, în urma experimentelor au fost obţinute rezultate mai puţin precise decât în cazul utilizării algoritmului Cuckoo Search, cu observaţia că o precizie bună a fost obţinută după un număr de evaluări ale funcţiei fitness de aproximativ trei ori mai mic. Având în vedere că evaluarea reprezintă cea mai costisitoare și consumatoare de resurse etapă a optimizării, considerăm că utilizarea algoritmului Fireworks poate fi mai avantajoasă. Legat de optimizarea multi-obiectiv, a fost propusă o strategie pentru creşterea numărului de soluţii Pareto optimale pentru un număr redus de iteraţii şi indivizi (fireworks) care participă la procesul de optimizare.

Cele 2 versiuni ale algoritmului FWA au fost implementate in C++, utilizând Microsoft Visual Studio 2015. Graficele au fost afişate folosind o aplicaţie Matlab în care au fost transferate rezultatele.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 3, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

IRIS RECOGNITION USING SPATIAL COLOR INDEXING

BY

IOAN PĂVĂLOI[*]

Romanian Academy – Iaşi Branch
Institute of Computer Science

Received: July 25, 2016
Accepted for publication: September 2, 2016

Abstract. An extension of a well-known simple global color criterion has been considered in this study. It was tested in image recognition on two known iris databases, UBIRIS and UPOL. Tests were made for three different color spaces, RGB, HSV and LAB using two discriminative classifiers, k-NN (k - Nearest Neighborhood) and SVM (Support Vector Machine). Three distances, Canberra, Euclidian and Manhattan were used in experiments. Two new distances were defined to be used with the feature vectors generated with the extension criterion. The results obtained using the extension color criterion are compared with the original one.

Keywords: iris identification; k-NN; SVM; color indexing; color features.

2010 Mathematics Subject Classification: 68M99, 68W35.

1. Introduction

Biometric recognition is at present a common and reliable way for person authentication. One of the most suitable, stable and reliable biometric technologies is iris recognition (Park *et al.*, 2011; Vatsa *et al.*, 2010), a method of identifying people based on unique patterns within the ring-shaped region surrounding the pupil of the eye. Iris recognition has some desirable properties

[*] Corresponding author; *e-mail*: ipavaloi@yahoo.com

such as uniqueness, stability, and non-invasiveness. The iris (which, in Greek, is the name of the goddess of the rainbow) is the annular region of the eye that is bounded by the pupil and the sclera. The iris is made almost entirely of connective tissue and smooth muscle fibres and it regulates the amount of light entering the eye. Iris colour is determined by the amount and the type of pigments present in the iris. Despite a common belief, actual change in colour of the iris rarely happens. While the colour of an eye may appear to change, this is typically due to lighting changes or perception based on nearby colours.

The identification process in iris recognition is based on gathering one (or, better, several) image of the eye using a high-resolution digital camera in visible or, typically, in near-infrared spectrum. Then, a specialized computer program can be used to compare the captured iris image with images stored in an iris database. Nowadays, researchers have attempted to perform iris recognition in the visible spectrum, numerous research projects being conducted in developing effective algorithms for iris segmentation and feature extraction. There are a lot of reasons that motivate researchers to direct their study focus to processing colour iris images (*i.e.*, RGB). We can briefly mention the interest in performing iris recognition on smart phones, using their colour cameras, and iris recognition at longer distances, 4 to 8 m (visible light can circumvent some of the problems associated with NIR imaging that requires the LEDs to be in close proximity to the ocular region) (Tankasala *et al*., 2012). So, iris recognition in the visible spectrum is an area of active research. Moreover, it has been shown that iris recognition can be improved using together NIR (near-infrared spectrum) and visible spectrum (Proenca, 2013; Santos *et al.*, 2010).

Since colour is an important feature which provides useful information, it can be used in image recognition for image indexing. Swain (Swain and Ballard, 1991) proposed a simple and effective image indexing method (named colour indexing) based on RGB colour histograms. This method has some very interesting properties, being insensitive to geometric transformation. Moreover, it is robust against variations such as shifting in viewing position or partial occlusion.

In this paper, we propose an extension of the known Swain colour indexing criterion. Using features related to the position of the colour pixels in the image, the performance of image recognition is improved. The experimental results in two public and well-known iris databases (UBIRIS and UPOL), demonstrate that our proposed method outperforms the original Swain method in terms of correct recognition rate.

In Section 2 we present the two iris databases on which we were worked on. Section 3 introduces the three colour spaces on which tests were made and the used distances. Section 4 mainly focuses on classifiers used and the process generating feature vectors. Experimental results and comparisons between the two criteria and discussion are presented in Section 5, respectively. Section 6 summaries this study and show the future work.

2. Iris Databases

This study was done on two well-known iris databases, UBIRIS and UPOL. UBIRIS is a dataset of 1.205 iris segmented images. The iris original images are part of the UBIRISv1 database (Proença and Alexandre, 2004; Proença and Alexandre, 2005) and their segmentation was performed automatically according to the details in (Radu *et al*., 2011). Each segmented iris image is rectangular with the same dimension of 360 ×100 pixels, coded RGB. In this dataset, there are 5 iris images for each of the 241 persons that participated to its development. Two images are considered similar if they belong to the same person.

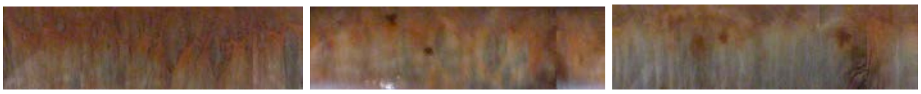


Fig. 1 – Examples of images in the UBIRIS database.

The UPOL iris database (Dobeš and Machala, 2005), contains 384 images belonging to 64 persons, three images for each eye. This dataset of high quality unsegmented iris images was made at the Palacký University, Olomouc (UPOL), the Czech Republic. The images, coded in PNG format are of good resolution (576×768), with 24 bits of RGB color for each pixel. The eyes were scanned using a TOPCON TRC50IA optical device connected with a SONY DXC-950P 3CCD camera (Dobeš *et al*., 2004; Dobeš *et al*., 2006; Dobeš and Machala, 2005). The images in this database are of very good quality, few images having defects like eyelids, eyelashes shadows or specular reflections.

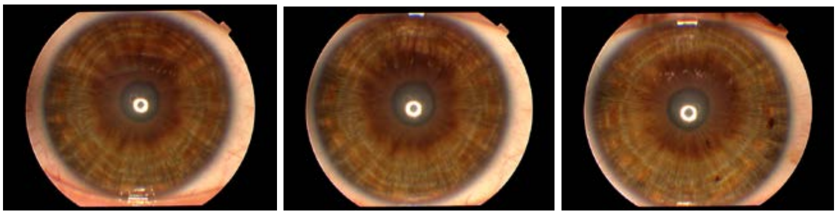


Fig. 2 – Examples of images in the UPOL database.

3. Color Spaces, Measures

Our study is made for three color spaces (Fairchild, 2005), RGB, LAB and HSV (which is also called HSB). The most commonly employed color space in computer technology is RGB (which stands for Red-Green-Blue). This color space is device-dependent and the representation of the colors in this space is adapted for monitors and cameras but difficult to understand intuitively.

For example, for color representation in user interfaces, other color spaces are usually preferred.

LAB is a uniform color space, in which same-size changes in the color coordinates also correspond to same-size recognizable changes in the visible color tones and color saturation (this is not the case with RGB color space). The international lighting commission CIE recommended the L*a*b*color space CZE 1976 and the L * u * v * color space CZE 1976 as an approximation of uniform color spaces (usually abbreviated as CIELAB or LAB and respectively CIELUV). Both color spaces are derived from the XYZ standard color system. LAB colors are absolute and the LAB color space is the only way to communicate different colors across different devices. Modifications to the CIELAB color space were continually developed to improve of the uniformity of the color appearance (for example LLAB color space).

HSV and HSZ (named perception-based color spaces), based intuitively on human color perception, are of interest for the fields of computer vision and computer graphics. In the HSV color space, Hue (H), saturation (S), and brightness value (V) are used as coordinate axes. A color can be more easily described intuitively (above all, by untrained users) by values for hue, color saturation, and intensity (than, for example, from vector components in the *RGB* color space).

Due to the fact that *RGB* space is regarded as an Euclidean space, known distance measurements can be adopted for the calculation of a color difference. In our study we have selected three measures to compute differences between feature vectors, Euclidian (1), Manhattan (2) and Canberra (3). These distances will be used in k-NN classification.

$$d_M(x, y) = \sum_{i=1}^{m} |x_i - y_i| \tag{1}$$

$$d_E(x, y) = \sqrt{\sum_{i=1}^{m} (y_i - x_i)^2} \tag{2}$$

$$d_C(x, y) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|} \tag{3}$$

4. Classifiers, Feature Extraction

We have chosen to use for this study two well-known discriminative classifiers, k-NN and SVM (Chang and Lin, 2013). To estimate the error rate of the k-NN classifier we used a validation technique named Leave-one-out cross validation (LOO-CV), a special case of cross-validation, in which the test subset is made of a single vector, all the other vectors being used as a training set (Dougherty, 2013). This procedure is repeated *m* times (*m* being the size of the dataset). So, this method uses as much data as possible for training, each vector of the dataset being used for testing exactly once. The total error is obtained by averaging the errors for all the runs. Of course, this type of validation is computationally quite expensive (evidently, more expensive than other methods), but it is very useful when the most accurate estimate of a classifier's error rate is required. For SVM, we generate randomly sets of tests containing exactly one iris for each person, the remaining ones from the dataset being used for training. The total error is obtained by averaging the errors for a number of runs.

We should first consider the Swain and Ballard color indexing scheme, in which, the range of the data for the color information of an image is split (divided) into "n" equally sized bins. For each bin the number of points from the data set (here, the colors of the pixels in an image) that fall into it are counted to obtain the feature vector $X = (x_1, x_2, \ldots, x_m)$, $m = n \times n \times n$. For example, in the RGB color space, if we divide each channel R, G, and B into 4 equally intervals with a length of 64 (0-63, 64-127, …, 192-255), we'll get a vector with 64 features. The images to be compared should have the same number of partitions with the same size. Moreover, they should have the same number of pixels, otherwise normalization will be required. The feature vectors for a division is formed from the set of all vectors generated for all images in the dataset.

In the extension criterion we propose, we also take into consideration the position for each pixels and we shall compute the average position for all pixels in each bin. As a result, we shall get other two vectors, in an "*m*" dimensional space, $m = n \times n \times n$:

- $(P_1, P_2, P_3, \ldots, P_m)$
- $(Q_1, Q_2, Q_3, \ldots, Q_m)$

where (P_i, Q_i) is the average position for "*i*" bin. For the extension criterion, a feature vector for an image is obtained concatenating the features obtained as for Swain and Ballard color indexing scheme with the two 3*n-dimensional vectors described above, $X = (x_1, x_2, \ldots, x_m, P_1, P_2, P_3, \ldots, P_m, Q_1, Q_2, Q_3, \ldots, Q_m)$.

For these vectors we define also other two measures, a "Manhattan" like measure d_{PM} and an "Euclidian" like measure d_{PE}. We also tested some Canberra like measures, but the results were not very good. Considering two vectors, $X = (x_1, x_2, \ldots, x_m, P_1, P_2, P_3, \ldots, P_m, Q_1, Q_2, Q_3, \ldots, Q_m)$ and $Y = (y_1, y_2, \ldots, y_m, R_1, R_2, \ldots, R_m, S_1, S_2, \ldots, S_m)$:

$$d_{PM}\left(x, y\right) = \sum_{i=1}^{m} \left|x_i - y_i\right| * \left(\left|P_i - R_i\right| + \left|Q_i - S_2\right|\right) \quad (4)$$

$$d_{PE}\left(x,y\right)=\sqrt{\sum_{i=1}^{m}\left|x_i-y_i\right|^2*\left(\left|P_i-R_i\right|^2+\left|Q_i-S_i\right|^2\right)} \qquad (5)$$

5. Results

Considering the iris recognition problem on two sets of feature vectors, the first one, named O-SET (Original Set) generated using Swain indexing color scheme and the second one, named E-SET (Extended Set), generated using the extension we propose, we'll analyze the results obtained using the two discriminative classifiers, k-NN and SVM on UBIRIS and UPOL datasets. The tests were made for three color spaces, LAB, RGB and HSV. In the next tables L stand for LAB, R for RGB and H for HSV color space, M for Manhattan, E for Euclidian and C for Canberra distance. So, for example HM expresses HSV color space and Manhattan distance. In the tests done we used k = 1 and k = 3 for the k-NN classifier. Due to the fact that there are a few iris images for each person, the best results are obtained for k = 1.

5.1. Results on O-SET Set of Feature Vectors for UBIRIS and UPOL Datasets

The recognition results (in percent's) on O-SET obtained using k - NN for k = 1 and using three distances, Manhattan, Euclidian and Canberra are presented in Table 1 for the UBIRIS dataset and in Table 2 for the UPOL dataset.

Table 1
1-NN Results on O-SET for UBIRIS Database

Division	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
LC	22.57	76.02	76.93	78.42	87.72	86.64	89.63	90.79	91.29
LE	12.45	58.42	57.01	63.57	86.97	71.87	92.20	85.31	92.61
LM	12.61	63.15	62.16	67.88	89.21	78.51	93.44	88.96	93.86
HC	82.82	87.97	87.80	90.71	89.96	90.95	91.04	91.87	92.28
HE	84.40	89.63	90.79	92.53	92.70	93.69	93.11	94.19	94.52
HM	86.80	91.62	92.28	93.78	94.11	95.19	94.77	94.94	95.60
RC	71.20	81.99	88.71	90.12	91.37	92.28	93.53	94.02	94.61
RE	60.33	76.93	84.98	89.54	91.12	92.61	92.61	94.36	94.69
RM	64.23	80.17	86.80	92.12	93.28	94.02	94.44	95.60	96.18

Table 2
1-NN Results 1-NN on O-SET for UPOL Database

Division	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
LC	31.51	61.20	70.57	70.31	76.30	79.43	80.47	82.55	84.11
LE	41.15	58.59	90.63	69.53	91.41	88.02	91.93	91.93	94.01
LM	42.97	61.98	93.49	77.08	94.01	92.71	95.57	94.53	96.61
HC	77.60	84.38	87.76	88.02	89.84	91.15	91.67	92.97	93.75
HE	80.21	88.54	91.15	93.75	92.97	94.79	95.05	95.83	96.09
HM	85.16	92.45	95.31	96.35	96.35	96.61	97.40	97.66	98.44
RC	57.03	71.09	75.78	76.82	82.81	82.81	83.59	83.85	84.90
RE	76.82	84.11	89.58	92.71	92.97	93.23	95.83	95.57	96.35
RM	80.73	88.54	94.01	94.79	95.05	96.09	97.66	97.66	97.66

As it can be observed, for UBIRIS, the best result is 96.18% and for UPOL it is 98.44%. For the 4×4×4 division, that is most used in practical applications, the best result is 91.62% for UBIRIS and 92.45% for UPOL, the two values being obtained for Manhattan distance on HSV color space.

So, as can be seen in Fig. 3, for the UBIRIS dataset and in Fig. 4 for the UPOL dataset, using the Swain indexing color criterion on the three color spaces with Manhattan distance, the best results are obtained in the HSV color space. The results on RGB color space, for division 5×5×5 or higher are closed to the results obtained on HSV color space. The results on LAB color space are far away from the results obtained on the other two spaces, RGB and HSV for divisions like 3×3×3 or 4×4×4.

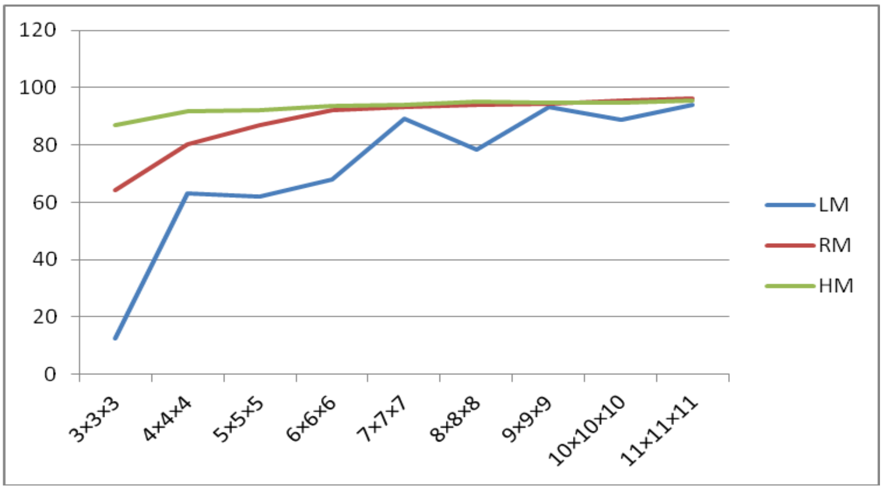


Fig. 3 – Results 1-NN on O-SET for the UBIRIS Database
using the Manhattan distance.

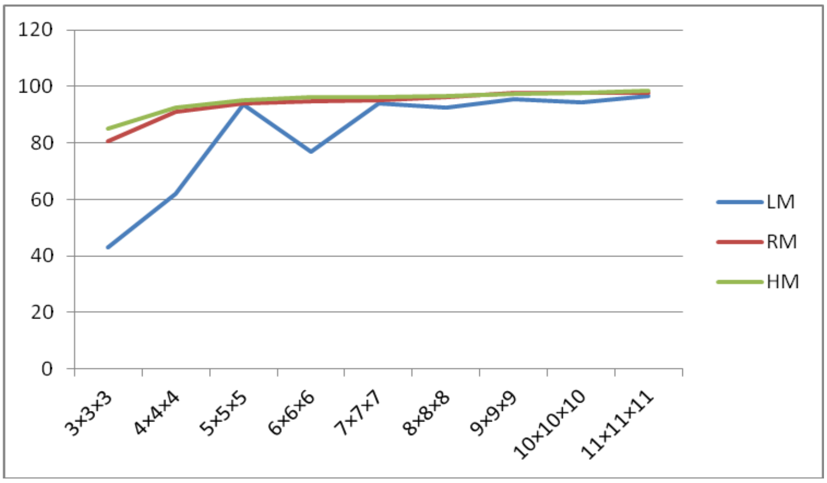


Fig. 4 – Results 1-NN on O-SET for the UPOL Database
using the Manhattan distance.

Considering the results obtained for UBIRIS and UPOL datasets, using HSV color space and Euclidian, Manhattan and Canberra distances, we notice by looking at Fig. 5 and Fig. 6 that the best results are obtained for Manhattan distance. This is also very convenient in practical application from the computational point of view. Nevertheless, in practical problems we certainly do not intend to use 11×11×11. So, the usage of Manhattan distance and 4×4×4 division on HSV color space seems to be a good choice for practical reasons.

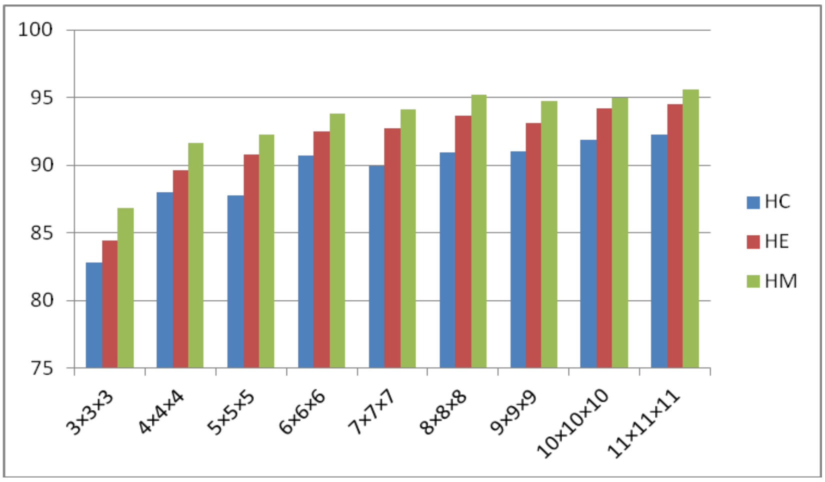


Fig. 5 – Results on O-SET for the UBIRIS Database using HSV colour space.

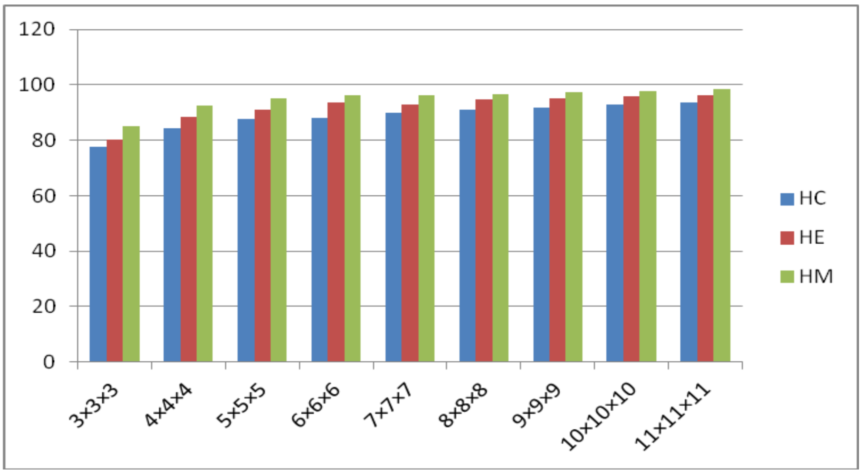


Fig. 6 – Results on O-SET for UPOL Database using HSV colour space.

The recognition results on the feature vectors obtained for UBIRIS and UPOL datasets using Swain color indexing scheme with the SVM classifier are presented in Table 3 and Table 4.

Table 3
Results SVM on O-SET for UBIRIS Database

	3	4	5	6	7	8	9	10	11
LAB	57.68	69.12	71.95	76.35	90.62	83.65	94.02	88.46	94.27
HSV	88.13	92.12	92.78	94.61	94.27	95.02	94.69	93.28	95.35
RGB	72.28	84.48	73.03	91.87	93.94	94.19	94.43	95.18	95.85

Table 4
Results SVM on O-SET for UPOL Database

	3	4	5	6	7	8	9	10	11
LAB	54.17	65.63	93.23	75.00	94.53	92.19	95.05	93.75	96.61
HSV	85.94	93.75	93.75	95.83	96.35	96.09	97.39	97.40	97.92
RGB	83.85	89.06	93.48	95.05	95.57	95.57	97.66	97.66	97.92

The best result using SVM on O-SET for the UBIRIS dataset is 95.85% and for the UPOL dataset it is 97.92%. As we can see, these are close to the greatest results obtained using k-NN, 96.18% and respectively 98.44%. As for the k-NN classification, the results obtained on HSV space are close to those obtained on RGB color space on each division and superior to those corresponding to the usage of LAB color space.

5.2. Results on E-SET Set of Feature Vectors for UBIRIS and UPOL Datasets

The recognition results on E-SET set of feature vectors obtained using k-NN for k = 1 and using three distances, Manhattan, Euclidian and Canberra are presented in Table 5 for the UBIRIS dataset and in Table 6 for the UPOL dataset.

Table 5
1-NN Results on E-SET for the UBIRIS Database

Division	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
LC	53.20	67.14	66.97	65.48	79.59	74.19	79.92	82.41	83.40
LE	20.75	65.81	60.00	68.22	87.63	76.35	92.45	86.56	92.86
LM	23.65	75.02	72.78	79.34	92.70	86.22	94.69	92.12	95.19
HC	75.27	79.09	79.09	83.32	84.48	86.89	87.14	88.05	89.05
HE	86.06	90.62	91.37	93.44	93.53	94.36	94.27	94.52	93.78
HM	90.95	94.11	93.61	94.36	94.44	94.61	94.69	94.69	94.61
RC	62.99	69.71	79.25	82.49	84.23	86.89	88.38	89.29	89.63
RE	62.99	79.25	85.81	90.54	91.45	92.86	93.20	94.44	95.02
RM	73.36	87.14	90.54	93.28	95.02	94.85	95.77	95.68	96.27

Table 6
1-NN Results on E-SET for the UPOL Database

Division	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
LC	25.26	50.52	54.17	59.11	63.80	67.45	69.53	70.05	72.40
LE	41.93	58.85	90.89	69.53	91.41	88.28	91.93	91.93	94.01
LM	47.14	67.45	93.75	81.51	94.53	93.75	96.09	95.57	96.35
HC	65.63	71.09	72.92	75.00	78.65	79.95	81.25	83.85	85.16
HE	80.21	89.58	91.41	94.01	93.49	95.05	95.31	95.83	96.61
HM	88.02	95.05	95.83	96.88	96.88	97.14	97.66	97.66	97.40
RC	49.74	56.77	66.67	65.63	71.35	70.83	72.92	72.92	75.52
RE	76.82	87.50	89.84	92.71	92.97	93.23	96.09	95.57	96.35
RM	81.25	91.15	94.27	94.79	95.83	96.35	96.88	96.35	96.35

The recognition results on E-SET for UBIRIS and UPOL datasets using the SVM classifier are close to the results on O-SET set of feature vectors.

Using the two distances we define in (4) and (5) we get the results in Table 7 and Table 8. First letter on first column on the table stands for the color space (L for LAB, H for HSV and R for RGB). The other two letter indicate the distance, PE for d_{PE} and respectively PM for d_{PM} distance.

Table 7
1-NN Results on E-SET UBIRIS Database

Division	3	4	5	6	7	8	9	10	11
L-PE	48.30	85.39	86.56	88.80	94.61	90.95	95.35	92.95	95.68
L-PM	50.37	87.97	89.13	90.79	95.60	92.45	96.35	94.61	96.18
H-PE	93.03	95.19	94.69	95.52	95.93	96.18	96.35	96.35	96.60
H-PM	94.27	95.93	95.35	96.18	96.18	96.35	96.68	96.76	96.85
R-PE	86.80	92.61	92.20	95.27	95.10	95.60	95.27	95.60	95.93
R-PM	89.21	93.94	94.44	95.77	96.02	96.18	96.27	96.35	97.10

Table 8
1-NN Results on E-SET on UPOL Database

Division	3	4	5	6	7	8	9	10	11
L-PE	54.43	73.18	88.02	80.21	88.02	92.45	91.41	90.63	93.23
L-PM	60.94	78.13	90.10	87.76	92.71	94.79	94.53	94.79	95.83
H-PE	83.59	89.58	90.89	92.19	92.45	94.01	93.75	94.53	95.57
H-PM	89.84	93.75	94.79	96.61	96.35	96.88	96.61	97.66	98.18
R-PE	80.47	87.50	90.10	90.36	92.71	92.71	94.27	93.49	94.79
R-PM	84.11	90.89	92.19	94.53	95.05	95.31	96.61	96.61	96.88

Comparing the results obtained on Swain color indexing and the extension, for HSV and Manhattan distance we have for UBIRIS dataset (Fig. 7):

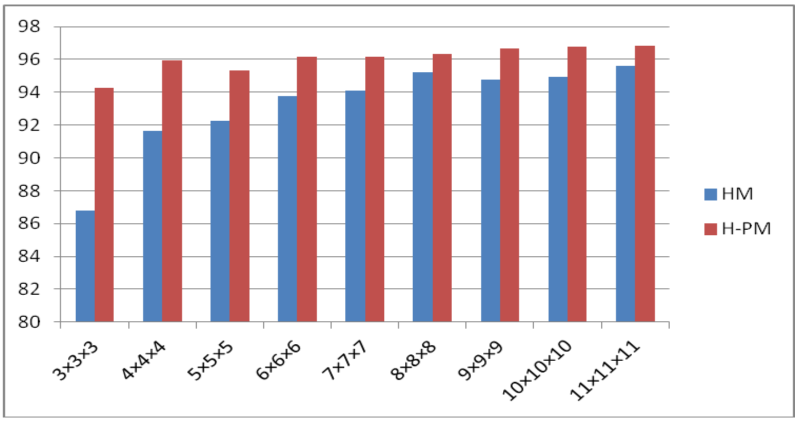


Fig. 7 – 1NN results for HSV space on UBIRIS dataset.

As it can be seen, the results obtained using the extended criterion are better than those obtained using the original criterion, for all divisions tested. The same results we obtained for UPOL dataset.

Studying the results obtained for UPOL dataset using O-SET and Manhattan distance and E-SET and Manhattan distance and for UPOL, comparing for example the results the results obtained on 3×3×3 and 4×4×4 with the original criterion (OC) and the extended criterion (EC), Fig. 8, it can be observed that the results obtained with the EC are better.

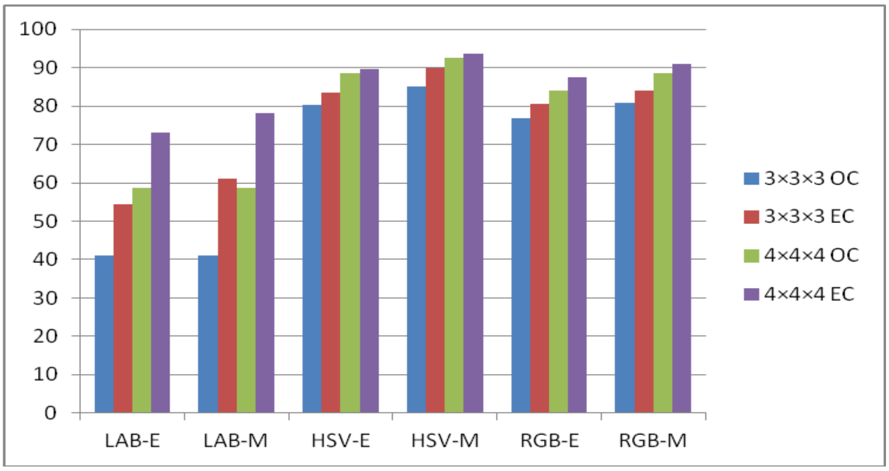


Fig. 8 – 1NN results on UPOL using OC and EC for 3×3×3 and 4×4×4 divisions.

Sure, we are not interested for divisions as 11×11×11 because feature vectors would have too many components. Sure, for more divisions, the result become closer to each other. In applications, for practical reasons, we are really interested in 3×3×3 or 4×4×4 divisions.

6. Conclusion, Future Work

In this study, part of the work in CBIR system building, we have considered an extension of a well-known simple global color criterion and we test it for image recognition on two iris databases. The test were made on three color spaces using two discriminative classifiers and three distances. Two new distances were defined to be used with the feature vectors generated using the extension criterion.

Studying and comparing the results obtained for divisions 3×3×3, 4×4×4, up to 11×11×11, we could conclude that the results obtained using the extension criterion are better. Even if the results for HSV and RGB are very close, the best results are obtained in the HSV color space. Moreover, for almost all the tests made, the usage of Manhattan distance gets better results than Euclidian and Canberra. Keeping in consideration the calculus complexity, Manhattan distance is a better choice comparing with Euclidian and Canberra distances. We have to notice that we have also tested other distances like

Chebyshev and Minkowski. The results obtained on UBIRIS database using the same dataset with optimal bin boundaries for the LAB color space using a 5×5×5 division were superior and reached a maximum percentage of recognition of 96.76 (Ciobanu *et al.*, 2013; Păvăloi *et al.*, 2013), being close to the results obtained using RGB color space and 10×10×10 equal division. We have to notice that the image dataset, being obtained using an automatic segmentation techniques, it is not perfect, hence the impossibility to attain 100% recognition. For UPOL, working on segmented images and with optimal bin boundaries for the LAB color space using a 5×5×5 division the maximum percentage reached was of 98.70% (Ciobanu *et al.*, 2014). In next steps we'll use this extension criterion on optimal bin boundaries for the HSV color space. Here, it's important to notice that on UPOL we worked on unsegmented images.

One advantage of this simple global color criterion extension is that we can add it or not in a CBIR system, depending on the user requests and other options (so, it's possible to use or not the extended set of vectors).As a conclusion, this simple global color criterion extension can be considered as being very good in practice and we we'll use as part of a CBIR system. We will consider it not only for iris images databases, but also for skin disease images databases. Sure, for practical reasons it is better to also take the texture into consideration and that will be one of our future works.

REFERENCES

Balas V.E., Fodor J., Várkonyi-Kóczy A.R., Dombi J., Jain L.C., *Soft Computing Applications*, Springer-Verlag, Berlin Heidelberg, 2013.

Chang C.-C., Lin C.-J., *LIBSVM: A Library for Support Vector Machines*, version 2.3, 2013.

Ciobanu A., Luca M., Păvăloi I., Barbu T., *Iris Identification Based on Optimized LAB Histograms Applied to Iris Partitions*, Bul. Inst. Polit. Iaşi, **LX (LXIV)**, *1*, 2014.

Ciobanu A., Costin M., Barbu T., *Image Categorization Based on Computationally Economic LAB Color Features*, Advances in Intelligent Systems and Computing, **195**, pp. 585-593, 2013.

Dobeš M., Machala L., Tichavský P., Pospíšil J., *Human Eye Iris Recognition Using the Mutual Information*, Optik, **115**, *9*, Elsevier, 399-405, 2004.

Dobeš M., Machala L., *Iris Database*, http://www.inf.upol.cz/iris/ UPOL Iris Database, 2005.

Dobeš M., Martinek J., Skoupil D., Dobešová Z., Pospíšil J., *Human Eye Localization Using the Modified Hough Transform*, Optik, **117**, *10*, Elsevier, 468-473, 2006.

Dougherty G., *Pattern Recognition and Classification - An Introduction*, Springer Science+Business Media New York, ISBN 978-1-4614-5322-2, 2013.

Fairchild M.D., *Color and Image Appearance Models*, Color Appearance Models, pp. 340-385, John Wiley and Sons, 2005.

Park U., Jillela R., Ross A., Jain A., *Periocular Biometrics in the Visible Spectrum*, IEEE Transactions on Information Forensics and Security (TIFS), **6**, 96-106, March, 2011.

Păvăloi I., Ciobanu A., Luca M., *Iris Classification Using WinICC and LAB Color Features*, The 4th IEEE International Conference on e-Health and Bioengineering - EHB 2013, Iași, November 21-23, 2013.

Proença H., Alexandre L.A., *UBIRIS - Noisy Visible Wavelength Iris Image Databases*, http://iris.di.ubi.pt/ubiris1.html 2004.

Proença H., Alexandre L., *UBIRIS: A Noisy Iris Image Database,* In Image Analysis and Processing – ICIAP 2005, Vol. **3617**, F. Roli and S. Vitulano (Eds.), Springer Verlag Berlin Heidelberg, pp. 970-977, 2005.

Proença H., *Handbook of Iris Recognition,* Ch. Iris Recognition in the Visible Wavelength, 151-171, Springer, 2013.

Radu P., Sirlantzis K., Howells W.G.J., Hoque S., Deravi F., *A Versatile Iris Segmentation Algorithm,* Presented at the BIOSIG 2011, Darmstadt, Germany, 2011.

Santos G., Bernardo M.V., Proença H., Fiadeiro P.T., *Iris Recognition: Preliminary Assessment About the Discriminating Capacity of Visible Wavelength Data*, IEEE International Symposium on Multimedia (ISM), pp. 324-329, 2010.

Swain M.J., Ballard D.H., *Color Indexing,* International Journal of Computer Vision, **7**, *1*, 11-32, 1991.

Tankasala S.P., Gottemukkula V., Saripalle S.K., Nalamati V.G., Derakhshani R., Pasula R., Ross A., *A Video-Based Hyper-Focal Imaging Method for Iris Recognition in the Visible Spectrum*, IEEE Conference on Technologies for Homeland Security (HST), pp. 214-219, 2012.

Vatsa M., Singh R., Ross A., Noore A., *Quality-Based Fusion for Multichannel Iris Recognition*, The 20th IEEE International Conference on Pattern Recognition (ICPR), pp. 1314-1317, 2010.

IDENTIFICAREA IRISULUI UMAN FOLOSIND INDEXAREA SPAŢIALĂ A CULORII

(Rezumat)

Lucrarea propune o extensie a metodei propusă de Swain și Ballard, metodă ce se utilizează ulterior în identificarea irisului uman folosind distribuţia locală a culorii pe suprafaţa acestuia. Testele s-au efectuat utilizând două baze bine cunoscute de imagini, conţinând fiecare un număr de imagini ale irisului pentru fiecare persoană. Prima din aceste baze de date este UBIRIS, ce conţine câte cinci imagini pentru 241 de persoane, un total de 1205 imagini. Este o bază de imagini de irişi, parte a bazei de date UBIRISv1, segmentarea realizând-se automat. Cea de-a doua bază de date folosită, de o calitate superioară a imaginilor, UPOL, conţine câte trei imagini pentru fiecare iris, stâng şi respectiv drept, pentru 64 persoane. În testele efectuate s-au utilizat doi algoritmi de clasificare discriminativi, k-NN și SVM. S-au utilizat trei distanţe clasice, Manhattan, Euclidiana şi Canberra. De asemeni s-au definit două distanţe particulare, corespunzătoare pentru metoda extinsă. Pentru caracterizarea distribuției locale a culorii într-o partiţie a irisului s-a realizat extragerea caracteristicilor de culoare pe spaţiile de culoare RGB, HSV şi LAB. Pentru aceste spaţii s-au realizat teste pentru diviziuni de la 3×3×3 la 11×11×11. Folosind metoda clasică propusă de Swain și Ballard, ratele de identificare pentru diviziunea 4×4×4 ajung la 91.62% pe baza de date UBIRIS şi 92.45% pe baza de date UPOL. În schimb, folosind metoda extinsă propusă de noi, ratele de identificare pentru diviziunea 4×4×4 ajung la 95.93% pe baza de date UBIRIS şi 93.75% pe baza de date UPOL. Metoda extinsă prezentată are avantajele unui volum computațional redus şi poate fi utilizată şi în aplicaţii de regăsire a imaginilor similare. Prezentul studiu face parte dintr-un proiect mai amplu în care dorim să realizăm un sistem de găsire a imaginilor similare într-o bază de date de imagini medicale.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 3, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

CONTOUR AND CENTERLINE TRACKING OF VESSELS FROM ANGIOGRAMS USING THE CLASSICAL IMAGE PROCESSING TECHNIQUES

BY

IRINA ANDRA TACHE[*]

University Politehnica of Bucharest,
Department of Automatic Control and Systems Engineering

Received: July 29, 2016
Accepted for publication: September 5, 2016

Abstract. This article deals with the problem of vessel edge and centerline detection using classical image processing techniques due to their simpleness and easiness to be implemented. The method is divided into four steps: the vessel enhancement which implies a non-linear filtering proposed by Frangi, the thresholding using Otsu method and the contour detection using the Canny edge detector due to its good performances for the small vessels and the morphological skeletonisation. The algorithms are tested on real data collected from a cardiac catheterism laboratory and it is accurate for images with good spatial resolution (512*512). The output image can be used for further processing in order to find the vessel length or its radius.

Keywords: medical imaging; non-linear filtering; edge detection; skeletonisation.

2010 Mathematics Subject Classification: 68M99, 68W35.

[*]Corresponding author; *e-mail*: irina.tache@acse.pub.ro, irina.andra@gmail.com

1. Introduction

Angiography is a still a standard technique used to evaluate abnormalities occurred in blood vessels, providing images with good spatial and temporal resolutions.

The blood vessels enter in the category of soft tissues and have attenuation values close to water. Because blood cannot be distinguished from the surrounding soft tissue, a contrast agent is injected into the vessels in order to increase their visualization on a radiographic image. The contrast material used in angiography is an iodine based substance, because it has high atomic number and physical density and it is not toxic to the body, even if a large quantity is injected into the blood stream. The iodine concentration is adjusted in order to visually distinguish the blood vessels from their surroundings tissues.

A compromise between the quantity of the contrast agent administrated to the patient and the levels of radiation is made and this will limit the image resolutions.

The drawbacks of this clinical investigation procedure are: the invasiveness and the irradiation of the patient.

Usually, from this medical investigation the morphological information needs to be extracted.

Therefore, the processing of X-ray angiograms engaged a lot of scientists in the last decades in finding a quick and robust method to be implemented on medical equipment. A couple of algorithms are already available on angiographs for vessel contour detection and centerline.

The image quality is an important prerequisite and it is usually related to contrast which is judged as the difference in gray tones between the objects and their background. The better the contrast of the image is, the easier will be for the physician to assess the normal or the abnormal anatomical structures.

The vessels’ analysis from an image consists of an examination of their topology or geometry, the gray levels and the neighborhood information.

The delineation of blood vessels on angiograms is an essential step in the computer aided diagnosis of the vascular diseases, although this is a challenging task due to the complexity of the image artifacts. Image noises have Poisson or Gaussian distribution.

The key step in improving the medical images for a quick diagnostic is the segmentation process which can be regarded as pixels classification into two groups: the object and the background. This step is necessary for centerline and contour detection of the vessels. This target is not properly achieved if the noise is not correctly filtered in the original image.

For the coronary angiograms when the heart and respiration movements are important, the digital subtraction angiography could not be applied. An alternative for the noise reduction is the multiscale filtering which have the advantage of optimally linking the spatial-frequency domains. They are used to eliminate the background structures and to enhance the vessels, especially the smaller ones. The resulted image is thresholded and further used by the centerline and edge detection algorithms.

The paper is organised as following: in the part 2 the spatial processing algorithms are detailed, in the part 3 the proposed algorithm is described and in the part 4, the results are analysed based on real data. Finally, the conclusions and further research directions are given.

2. Classical Image Processing Techniques

The main mathematical operations applied to one dimensional signals are adapted for the two dimensional case.

The main tasks of the computer scientists and engineers are the noise removal and the feature extractions from images. Therefore, some different filtering operations were developed each one dedicated to solve specific problems on certain image types.

2.1. Morphological Operations

The morphological mathematics was first developed by Georges Matheron in 60's. It is based on set theory is dedicated to the extraction of the object shape, such as the contours and the skeletons.

The morphological operators imply window operations applied on black-white or grayscale images. The set of all white pixels into an image is the morphological description of that image (Gonzalez and Woods, 2007).

The basic operations are: erosion (the suppression of the pixels next to the object's border), dilation (the addition of pixels next to the border), closing (the filling of small spaces between the objects' borders), skeletonisation (offers a compact representation of the object), pruning algorithm (the removal of unwanted lines). They work fine for images with uniform contrast and without important noises. Still, they are time consuming and a parallelization of the algorithms can be found in (Bräunl, 2001).

2.2. Angiogram Filtering

Even if the numerical imaging has revolutionized the medical image acquisition, there are still susceptible of noises induced by the electronic image transmission chain, by the sensor, by the electromagnetic fields nearby the acquisition equipment, etc.

The difficulty to analyze the X-ray angiograms comes from the low image contrast due to emitting a minimum dose of radiation to the patient. The contrast resolution for the coronary case, it has 2^8different gray levels. In addition to this, the high contrast anatomical structures such as bones can create shadows on the image and make the vessels investigation even harder.

In Alejandro *et al*. (1998) it is mentioned that an enhancement of the angiograms is an essential step for arterial problems diagnosis, angioplasty and bypass surgeries.

The noise reduction or the enhancement of certain components in the image is made by filtering operations. The smoothing operations include the median filtering which reduces the noise by replacing the value of each pixel, by its median gray value of its neighborhood.

Usually, the noises have a Poisson or Gaussian distribution.

Gaussian filters can improve the image, if the form and size of the filter are set accordingly to the standard deviation (s) of the Gaussian noise.

For example, a noise with a standard deviation of 0.5 will imply a mask size of 6*s which gives 5. So a 5x5 mask is used, with its coefficients computed as in Eq. (1):

$$g(x,y) = \frac{1}{2\pi s^2} e^{\frac{-\left((x-x_c)^2+(y-y_c)^2\right)}{2s^2}} \tag{1}$$

where x_c and y_c are the coordinates of the mask centre.

A small mask can cause less blurring to the filtered image and it permits the tagging of sharp lines in opposition to a large one which increase the blurring effect and it is used for large edges detection.

All these techniques are useful for removing the noise, but in return the object contour can be affected. Because the contour is critical in the image processing, the filtering methods must preserve it as much as possible.

The vessel enhancement techniques can be used for background components removal and for enhancing the small vessels, with low contrast. There are nonlinear filtering methods, which deal with the problem of non-additive noises without a normal distribution. They search in the images the geometrical structures which are the best candidates to represent the vessels.

Considering the vessels as connected, tubular and linearly piecewise curvilinear orientations structures, some filtering techniques are specifically designed (Sato *et al*., 1998; Meijering *et al*., 1999; Zwet *et al*., 1995) for them. There are also other methods such as ridge detectors or morphological operations, but with more applicability in the retinal vessels. Due to the artifacts complexity found in the cardiac and cerebral X-ray imagistics, these algorithms are time consuming and fail to detect vessel area on low contrast images (Cao *et al*., 2005).

Between the Gabor filtering and Frangi vesselness filtering the second one is investigated into this paper due to its better performances in angiograms processing as it was concluded in (Tache, 2015). Both of them use the capacity of the Gaussian function to obtain iteratively the magnitudes and the directions of the vessel structure. In order to cope with the problem of vessels' diameter variation, a multivariate scale is included.

For avoiding the noise amplification, a prior median filtering can be done.

2.3. Skeletonisation

The topological skeletonisationis the image processing which frequently follows the segmentation step. It is a compact representation of the objects with line structures. Considering the object shape (F) and its maximal disk D which respects the inclusion from Eq. (2):

$$D \subseteq F \tag{2}$$

The shape skeleton – $s(D)$ will be the set of maximal disks centers - $A_r(D)$ with r the shape radius (Petrou and GarcíaSevilla, 2006) and mathematically expressed as:

$$s(D) = \cup A_r(D) \tag{3}$$

For example, the skeleton of the vessel segment is in fact its centerline, which is equidistant to both borders. It can be computed with morphological operations, but it is sensible to noises (Vertan, 1999). This happens partially due to the fact that these operations work fine only for images with a uniform contrast. Nevertheless, this can be corrected with the aid of some special designed structural elements (Petrou and GarcíaSevilla, 2006) and by pruning algorithm which is dedicated to remove the spurs (parasitic components) which are non-important in the overall shape of the object. The main idea is to remove all braches shorter than a pre-established number of pixels.

From the end points, a certain number of points (m) will be eliminated from each branch. A morphological dilatation is applied on the new end points with the aid of a structural element which depends on the number m, in order not to shorten the eventual main lines.

2.4. Centerline Extraction

This is an essential step in the 3D vessel reconstruction because it preserves its curvilinear shape and length.

The most important methods for accomplishing this operation are: the ridge detectors based on level set theory, morphological skeleton, contour pruning, fast matching and geometrical model based methods, filtering based methods and vessels' tracking. For improving the results, some morphological operations such as erosion or closing should be applied to the images before and/or after the centerline extraction.

Sang *et al*. (2004) has succeeded to extract the centerline from the cerebral blood vessels in poor contrast and noisy background, as for example in the cases when the digital subtraction angiography methods cannot be used. They used the Gabor filtering by choosing a large scale. Still, due to the complicated artifacts of the X-ray angiograms, it can fail to detect all the centerlines.

Another solution for the centerline detection is the morphological skeletonisation which exploits the vascular shape from where the connectivity map can be constructed.

The tracking methods use a model to track the vessels path starting from a seed point and choosing the following vessel candidate based on a set of attributes, such as, direction, width and center point. The algorithm inherits the control theory concepts adapted to digital imaging and a good review dedicated to biomedical applications can be found in (Acton and Ray, 2005).

An alternative of this algorithm could be the manual selection of the intermediary pixels of the centerline. A method was implemented in Matlab®, to interpolate these points by a cubic spline function. The model comprises piecewise polynomials of third-orderwhich must verify separately a set of n control points, given by the user. A simple tridiagonal system of $n-2$ equations is formed by choosing the boundary conditions from the zero setting of the second order derivative of each polynomial at their endpoints (Bartels *et al*., 1995).

2.5. Edge Detection

The edge or contour detection is an important image processing which can be regarded as a segmentation step. It is defined as the linear characteristic of the object which has at least one neighbour outside the object. Actually, it implies the detection of the discontinuities in the gray levels.

Canny (1986) proposed a method for image gradient computation in the neighbourhood pixels and its steps are presented in the followings. The Gaussian filtered image which is a slightly blurred version of the original image is computed. Two Sobel operators can approximate the image gradient and a searching for the local maximum admitted by the output magnitudes in the gradient direction is performed. For limiting the multiple maximum detections due to the existing noises, a hysteresis thresholding is applied for detecting only the strong edges and their low connected branches using the direction information. This algorithm is more robust to noises and can detect the low contrast edges.

3. The Proposed Algorithm

Firstly, the vessels are enhanced using two consecutive filtering methods:

1. Median filtering - for avoiding the noise amplification.

2. Frangi vesselness filtering - detects the pixels which are the best candidates to represent the vessel.

3. After processing the maximum magnitude response, the edge and centerline are detected and some further tracking algorithms can be applied.

4. The graph nodes are defined at image coordinates where the probabilities of vessel occurrence are high.

5. The shortest paths from an initial point to all nodes in the graph are found by the Dijkstra's method.

The algorithm steps are presented in the Fig 1.

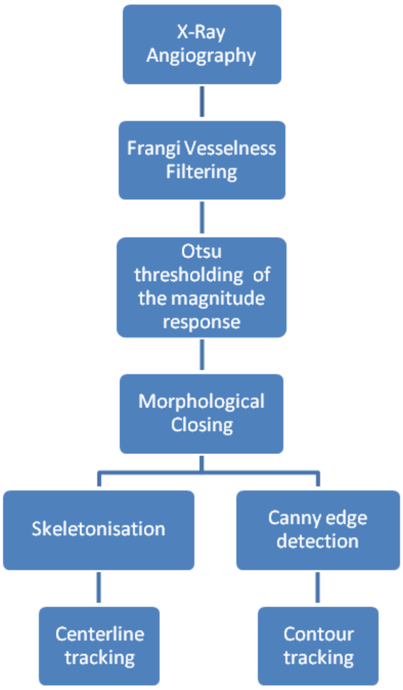


Fig. 1 – Image Processing Algorithm.

4. Results

The clinical protocol includes the acquisition of the patient fluoroscopic angiograms from different view angles and the EKG signal associated to each image set. The spatial resolution is 512*512 pixels and the temporal resolution is around 15 images/second.

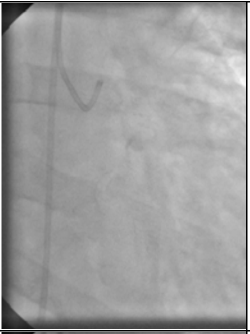
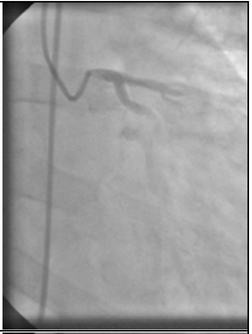
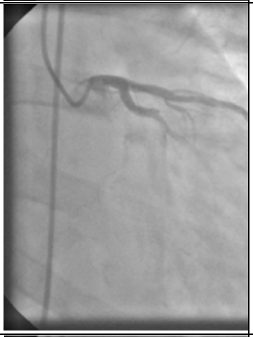
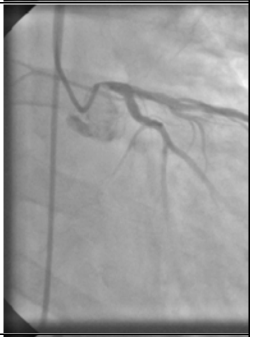
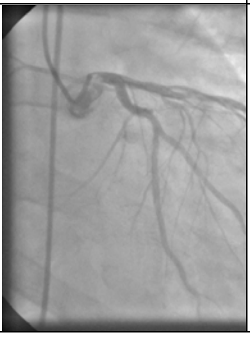
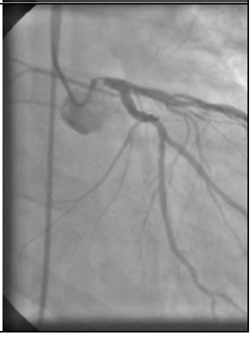
Images are stored in DICOM format, which is a set of standards for handling, storing and transmitting information in medical images. The file contains data about the acquisition setting of the medical equipment such as the rotation angles, the distances between the main components of the angiograph, the frames rate and the numerical image/images.

The coronary angiograms from 10 patients with the visualisation of the initial bolus in the first cardiac cycles were selected for study.

Some image samples are provided in Table 1 for evidentiate the progressively mixing of the contrast agent with the blood and its evolution throughout the cardiac cycle.

Table 1

A Sample of Coronarography Projections

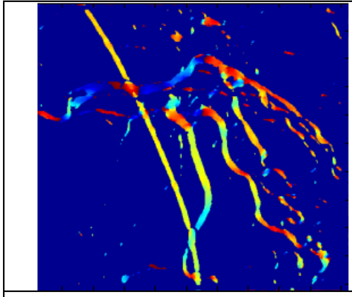
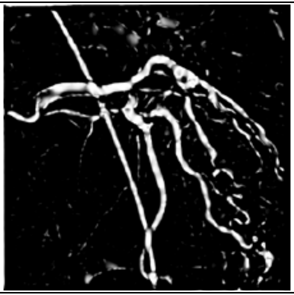
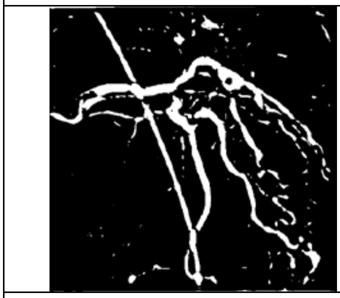
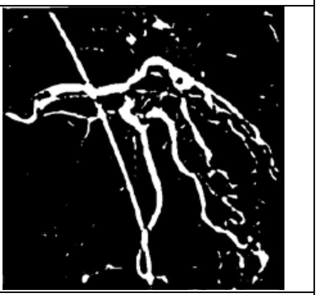
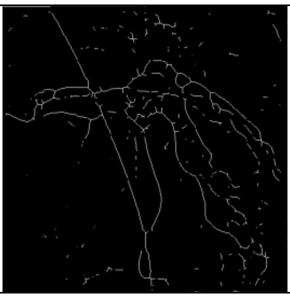
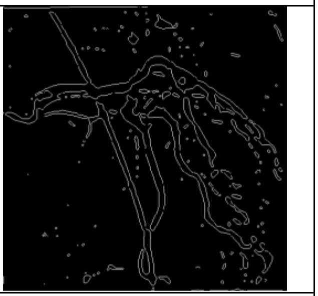
The movement artefacts must be avoided by choosing the images from the final diastole phase where the coronary vessels tend to remain motionless.

The image processing algorithm described in Fig. 1 was implemented in Matlab® and tested on a regular laptop with a medium frequency processor (around 2.6 GHz) with a minimum of 4 GB RAM.

The output images of the Frangi vesselness for a cardiac angiogram and their processing (thresholding, noise removal with morphological operations, skeletonisation and edge detection) are revealed in the Table 2.

In the Step 1 it is revealed a colour map with the direction information resulted from the Frangi vesselness filtering. In the Step 2, the magnitude response is shown and it is used for further processing.

Table 2
Edge Detection and Skeletonisation

	
1. Image with direction information	2. The magnitude response
	
3. Otsu thresholding	4. Morphological closing
	
5. Skeletonisation	6. Canny edge detection

Thirdly, the Otsu’s global thresholding is applied for obtaining a segmentation of the vessels. In the Step 4, a closing morphological operation is applied for filling the small gaps of the objects’ borders.

In the fifth step, the morphological skeletonisation is applied and in the sixth step the Canny algorithm is used for finding the vessels edges. In both cases there are residual spurs which must be eliminated with pruning operations.

Also, these results can be used for finding the graph nodes in order to construct the adjacency matrix of the pixels by searching the connecting components in the neighbourhood of the pixels. For defining the local neighbourhood connection, on the images from the steps 5 and 6 a filtering operation is applied using a window size of [5 5].

On the resulted images the minimal path cost of the nodes which are orthogonal to the tracking direction must be determined. This is accomplished with the well-known Dijkstra’s algorithm by using specific cost functions, such as, the distance between the nodes, their position, the information about the gray levels intensities and orientations extracted from the vesselness filtering. By minimizing the costs, a new path between the nodes/pixels is created and the edges are drawn in order to reconstruct the curvilinear shapes of the vessel.

Still, this method would not be completely independent of the human intervention, because it may imply the user selection of a start point on the vessel structure in order to construct the connectivity matrix. For relevant results, the above mentioned method requires images with good contrast.

5. Conclusions

For the coronary angiograms when the heart and the respiration movements are important, the digital subtraction angiography could not be applied. An alternative for the noise reduction is the multiscale filtering which have the advantage of optimally linking the spatial-frequency domains. They are used to eliminate the background structures and to enhance the vessels, especially the smaller ones. The resulted image is global thresholded and further used for centerline and edge detection.

The proposed algorithms are applied for real clinical data and their limitations are evidenced in the edge detection of vessels with high curvature.

The main goal of this spatial processing is to serve as an input for the minimal costs Dijkstra’s algorithm.

Sparse nodes are achieved by finding local maximum of Frangi vesselness map.

The method works well for all image sets and the vessel sizes were computed. Still the performance degrades when there is background clutter in the angiogram, such as spines, catheters, and guidewires.

Furthermore, vessel crossings could be incorrectly identified as vessel bifurcations.

The results can be used for measuring the 2D length and radius of a vessel’s segment selected by an image expert. Also, the centerline and the edges of the vessel segment can be used for the three dimensional reconstruction from two monoplane angiography.

Further work includes the refinement of the vessel structure extracted from a 2D angiogram by using prior shape information from a 3D coronary vessel model of the same patient.

Acknowledgments. This research was funded by the UEFISCDI National Project II for Partnership, Grant No. 130/29.07.2012 “High PErformance Computing of PersonAlized CaRdio ComponenT Models”.

REFERENCES

Acton S.T., Ray N., *Biomedical Image Analysis: Tracking*, Morgan & Claypool Publishers (2005).

Alejandro F.F., Niessen W.J., Vincken K.L., Viergever M.A., *Multiscale Vessel Enhacement Filtering,* Springer Verlag, pp. 130-137 (1998).

Bartels R.H., Beatty J.C., Barsky B.A., *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, The Morgan Kaufmann Series in Computer Graphics, San Francisco, pp. 9-17 (1995).

Bräunl T., *Tutorial in Data Parallel Image Processing*, Australian Journal of Intelligent Information Processing Systems (AJIIPS), vol. 6, no. 3, pp. 164-174 (11) (2001).

Canny J., *A Computational Approach to Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698 (1986).

Cao Z., Liu X., Peng B., Moon Y.S., *DSA Image Registration based on Multiscale Gabor Filters and Mutual Information*, IEEE International Conference on Information Acquisition, Hong Kong and Macau, China (2005).

Gonzalez R., Woods R., *Digital Image Processing*, 3rd Edition, Chapter 9, Pearson, Prentice Hall (2007).

Meijering E.H., Zuiderveld K.J., Viergever M.A., *Image Registration for Digital Subtraction Angiography*, Int. J. Comput. Vis., vol. 31, pp. 227-246 (1999).

Petrou Maria, García Sevilla Pedro, *Image Processing Dealing with Texture*, John Wiley & Sons Publishing (2006).

Sang N., Tang Q., Liu X., Weng W., *Multiscale Centerline Extraction of Angiogram Vessels Using Gabor Filters*, First International Symposium, Computational and Information Science, Shanghai, China, pp. 570-575 (2004).

Sato Y., Nakajima S., Shiraga N., Atsumi H., Yoshida S., Koller T., Gerig G., Kikinis R., *3D Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images*, Medical Image Analysis, vol. 2, no. 2, pp. 143-168 (1998).

Tache I.A., *Vessels Enhancement in X-Ray Angiograms*, IEEE International Conference on e-Health and Bioengineering, Iași, 2015, DOI: 10.1109/EHB.2015.7391549.

van der Zwet P.M., Reiber J.H., *The Influence of Image Enhancement and Reconstruction on Quantitative Coronary Arteriography*, Int. J. Card. Imaging, vol. 11, pp. 211-21 (1995).

Vertan C., *Prelucrarea și Analiza Imaginilor*, Printech Ed., București (1999).

DETECTAREA CONTURULUI ȘI A LINIEI CENTRALE ALE VASELOR DE SÂNGE DIN ANGIOGRAFII FOLOSIND TEHNICILE CLASICE DE PROCESARE A IMAGINILOR

(Rezumat)

Acest articol tratează problema detecției conturului și liniei centrale ale vasului folosind tehnici clasice de procesare a imaginii datorită simplității lor și a vitezei de calcul. Metoda este împărțită în cinci etape :

1. Se filtrează imaginea cu un filtru median și apoi cu un filtru neliniar de tip Frangi, care va determina pixelii cu probabilitatea cea mai mare de a reprezenta vasele de sânge.

2. Se aplică operația morfologică de închidere.

3. Se binarizează imaginea folosind metoda de prag global a lui Otsu.

4. Fie se detectează conturul cu ajutorul detectorului de margini al lui Canny datorită bunelor sale performanțe pentru vasele mici, fie se aplică operația morfologică de scheletizare.

5. Se detectează nodurile grafului din imaginea de la pasul 4 și se aplică algoritmul de calcul al costului minim al lui Dijkstra.

Imaginea rezultată poate fi utilizată pentru o procesare ulterioară cu scopul găsirii lungimii și razei unui segment de vas selectat de un expert în imagini medicale.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volumul 62 (66), Numărul 3 2016

AUTOMATICĂ şi CALCULATOARE

SUMAR

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volume 62 (66), Number 3 2016

AUTOMATIC CONTROL and COMPUTER ENGINEERING

CONTENTS

.