Improved Bi-optimal Hybrid Approximation Algorithm for Monochrome Multitone Image Processing

Rudolf Neydorf
Albert Aghajanyan
Department of Software Computer Technology and Automated Systems
Don State Technical University
Rostov on Don, Russia
Emails: ran_pro@mail.ru, foralbert92@gmail.com

Dean Vucinic
Vesalius College (VeCo)
Vrije Universiteit Brussel (VUB)
Email: dean.vucinic@vub.ac.be
Faculty of Electrical Engineering, Computer Science and Information Technology (FERIT)
Josip Juraj Strossmayer University of Osijek
Email: dean.vucinic@ferit.hr

***Abstract*– The paper investigates image tones approximation algorithm for the multitone image processing, which applications examples are in Web development, compression algorithms, machine vision etc. It considers the Monochrome Multitone Image (MMI) approximation of the original palette to be replaced by a palette having significantly less number of tones. For solving such problems, the optimization strategy requires the approximation quality, which maximizes the tones reduction deviation between the original and the approximated images. In particular, such problems are effectively solved with the heuristic Evolutionarily Genetic Algorithms (EGA) fulfilling the required accuracy, while computational costs still remain significant. Thus, this research is focusing on the hybrid algorithm that is combining the heuristic algorithm, in order to provide suboptimal approximation quality, and the deterministic Algorithm of Local Discrete Optimization (ALDO) for finding the local extreme. EGA minimizes the local discrete optimization search area and ALDO guarantees to find the extreme within the search area. In conclusion, such hybrid algorithmic architecture enables the MMI bi-optimization approximation.**

Keywords-heuristic algorithms; evolutionarily-genetic algorithm; image approximation; optimization; hybridization; bi-optimization.

I. Introduction

Image transformation has a wide range of applications. Examples can be found in the art image processing, where the image transformation is used for simplifying the graphical complexity of images by minimizing their file size, which makes their processing faster. Other examples are the recognition algorithms used for finding Objects for Autonomous Navigation (OAN). The goal of this research is to reduce the image palette, which simplifies and makes more efficient the recognition process.

In Technical Sight Systems (TSS) [4], when tracing the safe autonomous navigation route, the encountered objects are recognized by finding their orientation in space, and respective forms. This objective motivates the usage of the Monochrome Multitone Images (MMI). The term «multitone» is introduced to represent the images characteristic, which are defined with pixels of the same color, where each of them has different brightness. Often, the image pixels have different tones of gray color, called improperly «black and white», thus the term «halftone» is found more appropriate to use.

The processing of the real images with a huge number of details is focusing on the raster graphics and this research addresses the Raster MMI (RMMI) approximation.

One of the main properties of any MMI is associated with the Tones Palette (TP). TP is defined as an integer vector of the image tones. The TP properties depend on the TP's size or length, and on the respective TP tones values.

The standard scales are based on the equal tones values distribution. For example, the standard 256-tones palette is defined as the vector: $(0,1,2,\ldots 255)^T$. The special purpose scales (which include approximation) can be characterized by a non-equal tones distribution and having different number of tones in the related TP.

The TP size depends on the application domain and the related tones distribution rules depend on the structure of the image tones.

This paper describes the algorithm that transforms the images with large TP size, where each tone is maximized to its respective nearest original, resulting in a significantly smaller TP size, by applying an appropriate tones reduction algorithm. In addition, the computational efficiency is taken into account in order for the solution to be computed as fast as possible.

The rest of the paper is structured as follows. Section II describes the investigation goal and shows the approximation example. Section III presents the fully explained mathematical base of the approximation algorithm. The optimization technology of the developed approximation algorithm, together with the computed experiments, as examples are presented in Section IV. Finally, the conclusions are given in Section V.

II. Problem Formulation

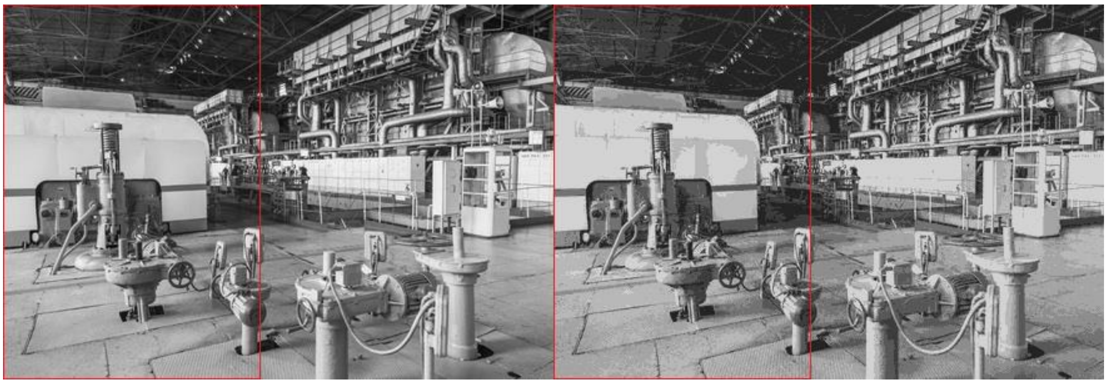
The main goal is to estimate the possibilities of the developed algorithm, which is designed to perform the optimal RMMI approximation, which transforms the Basic Palette (BP) of size n^0, to the RMMI Approximated Palette (AP) of size n^A, as fast as possible.

The optimal 8-tone TP, which approximated the original 256-tones image (Figure 1a), has the following tones distribution structure: (34, 55, 78, 107, 140, 165,

 ISBN: 978-1-61208-599-9

182, 205), as shown in Figure 1b. In Figure 2, the Brightness Frequency (BF) of original (on the top) and approximated image (on the bottom) are shown. The BF structure, shown at the bottom of Figure 2, shows the non-equal tones distribution. This optimum approximation distribution was obtained from the pixels` brightness non-equal tone distribution of the original image (Figure 2, at the bottom). By comparing the images in Figure 1a and 1b, even visually, it is quite easy to conclude that it is not always appropriate to use the big original palette size, as the obstacles recognition is based on the contours found from the image brightness change. Obviously, the application of such reduced palette size simplifies the pattern recognition tasks in such case.

However, it is also obvious that there are various algorithms that can provide such transformation and the difference between them can be found in their implementation complexity and the quality of approximation. The question to be answered is to validate the application of such optimal or suboptimal algorithms when reducing the RMMI palette size.



a) b)

Fig. 1. Original image with 256 tones palette (a) and transformed image approximated with 8 tones palette (b).

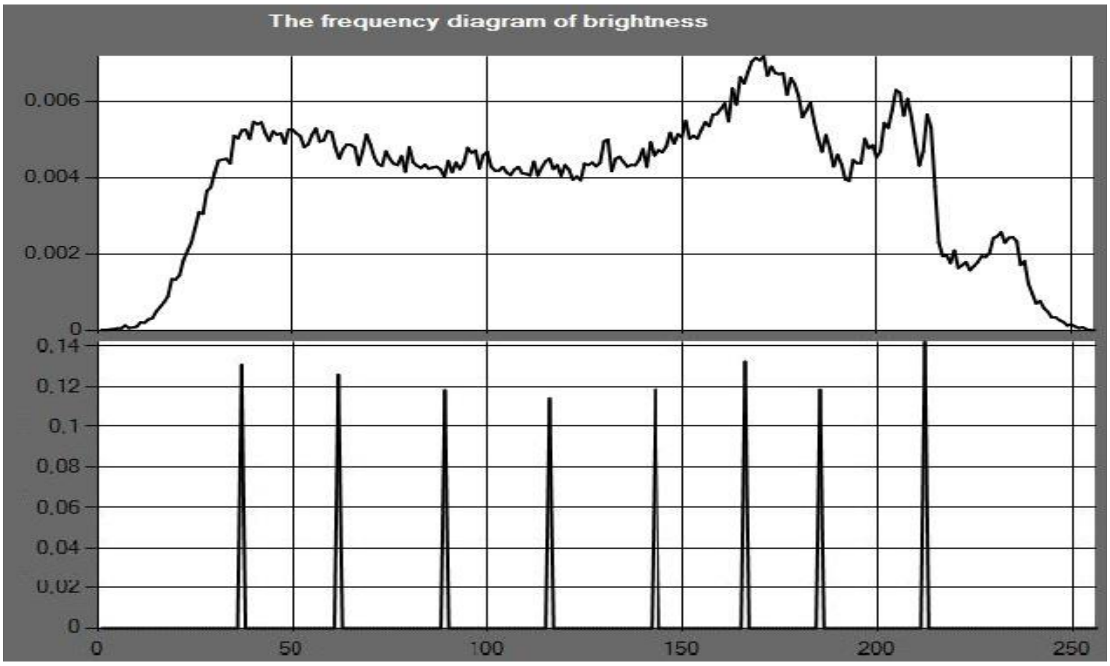


Fig. 2. The brightness diagram of original image (on the top) and the computed optimal brightness approximation (on the bottom)

The investigations have shown that the most effective way to solve the problem is to develop the sub-optimization algorithm, which takes into account the quality of the MMI approximation according to the complexity limits. In other words, the algorithm computing time should allow its practical application. Thus, the goal is to adopt such optimization search for the predefined conditions under the chosen quality criteria in order to achieve acceptable computing time to find extremes.

III. Mathematical Models And Algorithms Of Image Approximation Process

Mathematical model of RMMI. The raster image presents as array of pixels (p). In the rectangle image, the pixels are represented as an ordered array, which is defined by rows *i* and columns *j*

$$P = \{p_{ij} | \; i \in [1, r]; j \in [1, c]\}, \quad (1)$$

where r and c – number of rows and columns of pixels that are defined by certain *i* and *j* coordinates. The array P can be represented as a rectangle matrix with r rows and c columns:

$$P[i,j] = \begin{pmatrix} p_{11} & \cdots & p_{1i} & \cdots & p_{1r} & \cdots & p_{1c} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{i1} & \cdots & p_{ii} & \cdots & p_{ir} & \cdots & p_{ic} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ p_{r1} & \cdots & p_{ri} & \cdots & p_{rr} & \cdots & p_{rc} \end{pmatrix}, \quad (2)$$

where in (2) case c>r, which means that it is «widescreen image».

Some RMMI pixels have the same color, but different brightness, called tone. The RMMI palette is defined as the ordered array of (s) tones:

$$B^S = \{b^s | s \in [1, S]\} = (b^1, \dots b^s, \dots \; b^S). \quad (3)$$

where S – size of the palette, in other words, the number of tones that are used for image display.

Most tasks related to the recognition use RMMI. The ordinary RMMI uses 256 tones of gray color from 0 – absolute black, untill 255 – absolute white, and for each pixel represented with 8 bits ($2^8 = 256$).

Beside the raster image's pixels coordinates *i* and *j*, which are determine by (2), there is a need to add a brightness characteristic that define pixel tone. This pixel tone is defined by the index number of *s*, according to formula (3).

The RMMI's model quantity characteristic (2) is the brightness of pixel with use of its more convenient matrix, the Mathematical Model (MM) of image matrix, where each element is the pixel brightness. The position of each element in matrix (i, j) determines its position in the image rectangle and its number shows the brightness of the certain monochrome image pixel:

$$p_{irs} = b^s. \quad (4)$$

Then, the full MM of the raster monochrome image becomes appropriate to determine the 3-dimensional matrix (or array of 2-dimensional matrices (2) with 3th argument)

$$B^S[i,j,s] = \begin{pmatrix} b_{11}^s & \cdots & b_{1c}^s \\ \vdots & \ddots & \vdots \\ b_{r1}^s & \cdots & b_{rc}^s \end{pmatrix}, \; s = \overline{1, S}. \quad (5)$$

RMMI approximation scheme. The problem of raster image approximation is to reduce the image palette size. Thus, there is a need to decrease the number of tones, which are defining the image. The process consists in replacing the pixels brightness value of the Original Monochrome Multitone Image (OMMI), defined with the Original Palette (OP)

$$B^{S^0} = \{b_s^0 | s \in [1, S^0]\} = (b_1^0, \dots b_s^0, \dots \; b_{S^0}^0), \quad (6)$$

with the brightness values of the approximation palette, whose size is $S^a < S^0$:

$$B^{S^a} = \{b_s^a | s \in [1, S^a]\} = (b_1^a, \dots b_s^a, \dots \; b_{S^a}^a). \quad (7)$$

The original size of OP S^0 depends on many factors, but primarily is the number of details in the image. The size of the approximation palette S^a can be different and depends only on the approximated MMI application field. For example, in the MS Office Paint editor, only 3 types of colored palettes are available for processing colored image: «24-bit image» ($S=2^{24}$), «256-colors image» (S=256) and «16-colors image» (S=16). They all defined by formula (3).

The replacement process of image pixels palette (6) to pixels with smaller size palette (7) is called the tone approximation of RMMI. The OMMI transformation result is the Approximated Monochrome Multitone Image – AMMI.

The approximation can be based on significantly different algorithms. However, the underlying basis of all algorithms is the same and consist in changing the pixel tone from OP (6) to the reduced tone AP (7). Also, in OP - B^{S^0} we must define some subarrays of pixels $B_s^{S^0}$, which show the set of pixels that will be covered by one certain pixel from B^{S^a}:

$$B_s^{S^0} \subset B^{S^0} : \forall \; s \in [1, S^a] \rightarrow$$
$$\rightarrow \bigcup_{s=1}^{S^a} B_s^{S^0} = B^{S^0} \& \bigcap_{s=1}^{S^a} B_s^{S^0} = \emptyset \,. \quad (8)$$

The replacement mechanism of any pixel $b_s^0 \in B_s^{S^0}$ to one pixel of AP$b_s^a \in B^{S^a}$:

$$\forall s : b_s^0 \in B_s^{S^0} \rightarrow b_s^0 \cong b_s^a \in B^{S^a}. \quad (9)$$

Therefore, only 2 factors define the variation and the effectiveness of the algorithm that is converting OMMI to AMMI:

1. Structure of OP dividing on sub arrays $B_s^{S^0}$, in other words, how many and which pixels enter a subarray, where all elements changes their value to b_s^a;
2. Values of b_s^a, where every separate value equals to one of the elements of subarray $B^{S^0} \ni b_s^0 = b_s^a$, because the TP values are always natural numbers.

The result of processing a MMI image (4) with any approximation algorithm will result in AMMI, whose MM's structure will be different, because all pixels changed their tone value from OP (6) to smaller size AP (7). If the OMMI's model in palette (6) $P^{S^0}[i,j,s^0]$ contained pixels $p_{irs^0}, s^0 = \overline{1, S^0}$, then AMMI's model will contain $p_{irs^a},\ s^a = \overline{1, S^a}$. Thus, every pixel position in AMMI with coordinates (i,j) will be characterized by brightness error:

$$\Delta p_{ijs} = p_{ijs}^a - p_{ijs}^0, i = \overline{1,r}, j = \overline{1,c}. \qquad (10)$$

The total error of the image approximation should be calculated based on all AMMI pixels $N = r \cdot c$.

Estimation of monochrome multitone image approximation quality. Using the common form of writing (4) and its variants for OMMI and AMMI in brightness defined form (5) the error for all approximated pixels is defined by the following formula:

$$\Delta b_{ijs} = b_{ijs}^a - b_{ijs}^0, i = \overline{1,r}, j = \overline{1,c} \qquad (11)$$

where Δb_{ijs} – resulting deviation, P^A – matrix of approximated image, P^I – matrix of original image and $r \times c = N$ – number of all image pixels.

The difference between the estimation criteria depends on the applied formula for all the image pixels errors.

The investigated question in [13] was to define the most adequate criteria for image approximation. The result was to use the «Least Module of Deviation» (LMD):

$$\Delta_m = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\left|p_{ij}^A - p_{ij}^I\right| \qquad (12)$$

IV. The RMMI Approximation algorithms And AMMI Optimization

A. *RMMI approximation technology*

According to already considered information, the OMMI to AMMI approximation is the sequential execution of the following operations:

1) Chose the size of S^A AP (7);
2) Divide the OP (6) on S^A subarrays $B_s^{S^0}$ (8);
3) Chose S^A tones b_s^A AP from subarray $b_s^0 \in B^{S^0}$;
4) Replace sequentially OMMI $N = r \cdot c$ pixels that belong to OP with the nearest to them pixels from AP, which result is AMMI;
5) Evaluate the criteria (12) for AMMI.

To achieve the best possible approximation result, there is a need to organize the algorithm in a way that allows reaching the following condition:

$$\Delta_m\left(B^{S^0},\ B^{S^A}\right) = \min_{B^{S^a}} \Delta_m. \qquad (13)$$

Thus, it is necessary to realize the algorithm 1-5 as an algorithm for the minimization searching criteria (12). The high resolution of graphical objects (size of AP and number of pixels) motivates developing and using the heuristic Evolutionarily-Genetic Algorithm (EGA). The classic EGA modification on solving the approximation problem showed good accuracy [8], [11], [12] and [13].

B. *The evolutionarily-genetic optimization of AMMI*

It is obvious to consider the RMMI approximation technology (based on the biological mechanism of inheritance). Indeed, if the approximating pixel tones can be selected from OP (6), only then, this palette is equal to a genome, whose elements spawn any AMMI that represents an individual within the approximation algorithm. By following the above approximation technology steps, it is easy to conclude that AMMI-individual, as an image, undoubtedly is determined by the AP structure (7) and the dividing structure OP (7) form sub-arrays. This spawns some clear analogies: (1) between the AP B^{S^a} and chromosome, (2) between the tones b_s^a of AP and genes, (3) between pixels of OP $b_s^0 \in B_s^{S^0}$ from subarray, replaced by the tone b_s^a and alleles. All these analogies are providing the variability base for the spawned (by approximation) AMMI.

The first step of the algorithm is the creation of an initial population, which is the random variation of the chromosome alleles (pixels tones of OP within the defined ranges limits). This step allows creating different chromosomes. The individuals of the initial population are estimated according to the criteria from (12). The evaluation results are used to perform «roulette» selection of the population. After the mentioned selection, the population will change under the influence of the genetic operators «crossover» and «mutation», which, as a result, will create the new generation. The convergence of the proposed algorithm is attainment for the last generation.

The described EGA modification is well studied. The difficulties of the EGA application for solving the problems of the dynamic transformation and the image analysis are due to the insufficient computing speed. The needed speed up can be achieved by significantly reducing the population size and the respective number of generations, but, as a drawback, it will sharply decrease the accuracy and confidence level of the computed result.

Another drawback of EGA is the impossibility to estimate the nearness extreme to its optimum. This leads to the increase of the quantity parameters and the time needed to estimate the approximation quality. This motivates the idea to make hybridization between the algorithms of extreme estimation and EGA.

C. *The AMMI extreme estimation algorithm*

The developed deterministic algorithm for any chromosome extreme estimation is based on the genes that represented by natural numbers, which change only on the values that are divided by 1 and have no remainder in the result. This means that the neighborhood of the AP-chromosome (7), as defined at the point in S^a-dimensional space, will consist of the finite number of points, whose coordinates will be deviated from AP tones as 1, -1 or 0. This procedure is illustrated in Table 1.

The central column contains the point coordinates of the 8-dimensional space, for which the extreme has to be found. Any one-element combination in each Table 1 row, with the exception of the combinations of all the elements from the central column, gives a point coordinates in the neighborhood. It is easy to calculate the combinations number as $3^8 = 6561$.

TABLE 1. THE EXAMPLE OF NEIGHBORHOOD STRUCTURE OF INVESTIGATED SOLUTION

$b_{ij}-1$	$b_{ij}+0$	$b_{ij}+1$
32	33	34
52	53	54
73	74	75
98	99	100
131	132	133
160	161	162
179	180	181
204	205	206

All these points have to be checked by criteria (12) and, in addition, all the investigated AP points are checked with the condition:

$$\Delta_m\left(B^{S^0},\ B^{S^{A^{\{e\pm1\}}}}\right) > \Delta_m\left(B^{S^0},\ B^{S^{A^e}}\right), \quad (14)$$

where $B^{S^{A^e}}$ - the AP-chromosome that investigates on extreme, $B^{S^{A^{\{e\pm1\}}}}$ – the array of chromosomes, which are the nearest neighborhood of $B^{S^{A^e}}$. If the condition (14) is satisfied for all elements of array $B^{S^{A^{\{e\pm1\}}}}$, then the found AP solution is at least the local extreme. If there are some points that did not satisfy the extreme condition, then there is a possibility to choose a better point, and in that case, repeat the check. Such possibility motivated a deterministic algorithm creation, which seeks the real extreme in neighborhoods of that point.

It is obvious that the processing time of Extreme Estimation Algorithm (EEA) depends on the AP size. For example, for 8-tone AP, it is necessary to produce 6561 calculations of criteria (12) and check the condition (14). On a computer with the Intel quad 3.3 GHz processor these operations require ~23.85 s. But for 16-tone AP the number of combinations will increase to 3^{16} = 43046721, which means that the processing time will be 6561 times longer, requiring ~44 hours. Therefore, the application of EEA is only reasonable for small AP sizes. At the same time, the algorithm is quite appropriate for STS, which uses small size palettes.

In such case, it is possible to develop and use the Algorithm of Nearest Extreme Finding (ANEF). The algorithm is based on EEA, with the difference of having a new cycle starting condition, which will be initiated after finding the first AP that is better according to criteria (12) than the investigated one, but simultaneously it has not satisfied the condition (14). The stop condition ending the algorithm will be when the condition (14) is satisfied for the entire AP neighborhood. Therefore, ANEF can be also used as EEA.

Figure 3 shows the protocols of the extreme AP searching steps for the image fragment (Figure 1, displayed by red rectangle) using ANEF from different starting positions. As a result, it takes a different number of steps. However, the last part of the route is the same for all the cases, which can be seen according to equal time to the nearest AP search. In addition, the final extreme check step takes the same time (the small difference is explained by the processor background activity). If the EGA stage is excluded and only ANEF is used as start for the so-called "weighted" distribution [11], [12], then the processing time is 148.7 s.

However, to plan the searching strategy based only on ANEF is not reasonable, because even in this example, it is easy to notice that it depends on the starting position and the computing time can takes 1.5-3 minutes (as shown in this considered case). But, if the AP is provided with criteria value, for example, 415221 (Figure 3), then the time to find the optimum solution will decrease to ~36 s.

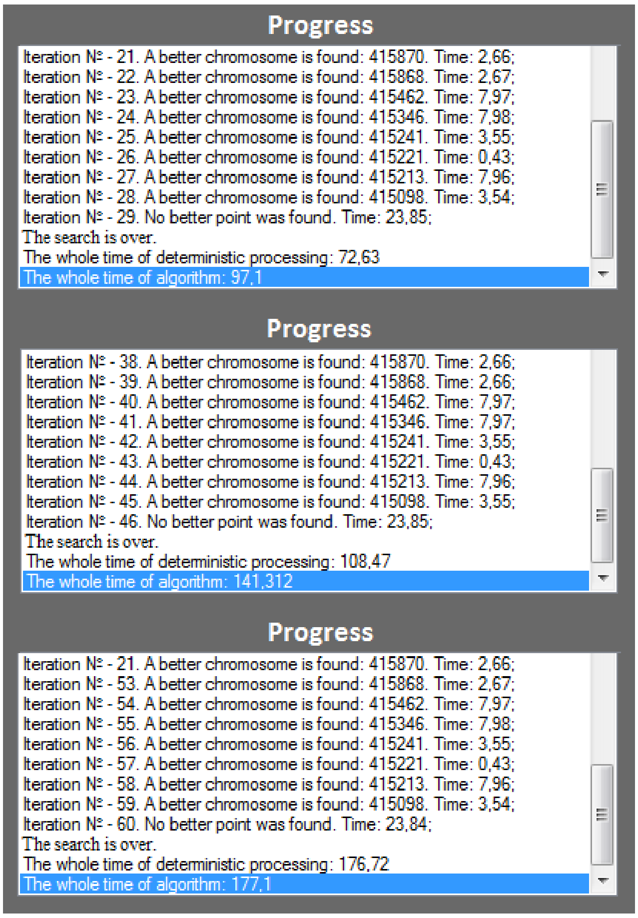


Fig. 3. The ANEF results of the extreme AP search with different starting positions

Therefore, the fair EGA initial setup is to provide a position near to extreme, and to further on apply ANEF, which will check for the solution significantly faster.

For the above-mentioned reasons, it is proposed to use Hybrid Extreme Search Algorithm (HESA), which is based on sequential EGA and ANEF. The results of this research are described below.

D. *The optimization of HESA*

For solving the HESA problem, it is necessary to find such EGA parameters that can find the solution in order for entering the extreme area to be realized within minimal time. These parameters include the number of parallel runs of EGA x_1, size of population x_2 and number of generations x_3.

The 2 Full Factor Experiments (FFE) have been planned and implemented. In the first FFE, the factors variation was on the same low level affecting all the factors: $x_1, x_2, x_3 \in \{2, 4, 6\}$. The results of FFE showed minimal average time (60.7 s.), when all x_i factors had high values. The regression coefficients values were negative for all the linear effects $b_1 = -15.3$; $b_2 = -8.3$; $b_3 = -11.6$.

Because the best FFE average time was too long and the increment observed for all the 3 factors was expected to give significantly better result. The second FFE was implemented with $x_1, x_2, x_3 \in \{4,6,8\}$. And, again, the best average time (47.0 s.) was obtained, when all factors have their maximal values. The regression coefficients give bigger values for the linear effects $b_1 = -18.1$; $b_2 = -17.6$; $b_3 = -12.9$.

The best result of the second FFE is quite near to the maximum faster time of HAES – 23.8 s, which gives the basis for a new investigation – to consider the gradient descent effect. The best result showed that the following parameters $x_1 = 12$; $x_2 = 12$; $x_3 = 10$, give an average time of 40.7 s to obtain the solution. The further EGA's parameters increment increased the computational time, but simultaneously the output result was much nearer to extreme according to criteria (12). The explanation can be found in the EGA processing time. For example, when setting $x_1 = 16$; $x_2 = 20$; $x_3 = 16$, the computational time to find the extreme becomes 544 s.

To summarize, and in accordance with the investigated part of the image, see Figure 1, the most effective solution of the problem was achieved when applying the hybrid algorithm, which is based on the sequential EGA application, with defined parameters, and algorithm of nearest extreme finding.

V. CONCLUSION AND FUTURE WORK

The main research results are:

1. The new developed deterministic Algorithm of Nearest Extreme Finding (ANEF), together with the extreme solution estimation showed to be highly efficient and, thus, it is found to be a very good tool for solving monochrome multitone image tones approximation problems.

2. ANEF became the basis for the hybrid extreme search algorithm, allowing a significant increase in the performance of MMI TP approximation. It enabled the approximation bi-optimization providing the quality estimation with respect to the required computational time.

3. ANEF is the NP full and thus not appropriate for solving the approximation problem, in case of the large TP size needs, which motivates future studies to investigate the possibilities on how to increase the ANEF computational performance.

REFERENCES

[1] J. Puzicha, M. Held, J. Ketterer, "On spatial quantization of color images", *IEEE Transaction on Image Processing*, vol. 9. no. 4, pp. 66-82, 2000.

[2] C. Emre, "Improving the Performance of K-Means for Color Quantization", *Image and Vision Computing*, vol. 29, pp. 260-271, 2011.

[3] Color quantization URL: wikipedia.org/wiki/Color_quantization (date of access: 9.02.2017)

[4] Bishop Ch., "Pattern Recognition and Machine Learning". *Springe*, 2006.

[5] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern classification*, Wiley, New York. 2001.

[6] D. Chirov, O. Chertova, T. Potapchuk, "Methods of study requirements for the complex robotic vision system", *Spiiran proceedings*, vol. 2(51), pp. 152-176, 2017.

[7] A. Aghajanyan, R. Neydorf, "Optimization of Monochrome Multitone Images approximation based on evolutionarily algorithm", *Omega Science*, Vol.108, pp.11, 2016.

[8] A. Aghajanyan, R. Neydorf, "Optimal approximation of monochrome multi-tone images using the evolutionarily genetic algorithm", *Com-Tech 2016*, pp. 108-112, 2016.

[9] R. Neydorf, A. Derevyankina, "Solving the multiextremal problems with particle swarm method", *Vestnik DSTU*, Vol. 4 (47), 2010.

[10] R. Neydorf, A. Derevyankina, "Solving the recognition problems with particle swarm method", *Izvestiya SFedU*, Vol.7 (108), 2010.

[11] R. Neydorf, A. Aghajanyan, D. Vucinic, "Monochrome Multitone Image Approximation on Lowered Dimension Palette with Sub-optimization Method based on Genetic Algorithm", *ACE-X 2016*, Split (Croatia), 2016.

[12] R. Neydorf, A. Aghajanyan, D. Vucinic, "Monochrome multitone image approximation with low-dimensional palette", *IEEE East-West Design & Test Symposium (EWDTS)*, 2016.

[13] R. Neydorf, A. Aghajanyan, "The research of the application possibilities of tones approximation in a technical vision for the autonomous navigation objects", *Izvestiya SFEDU*, Technical sciences, №1-2 (186-187), pp. 133-145, 2017.

[14] C. Pierre, R. Jean-Philippe, *Stochastic Optimization Algorithms*, Handbook of Research on Nature Inspired Computing for Economics and Management Hershey, 2006.

[15] Vinogradov I., *Mathematical encyclopedia*, Soviet encyclopedia, 1977.