About Computer Vision using Optimal Image Approximations

Mikhail Kharinov
Laboratory of Big Data Technologies
for Sociocyberphysical Systems
St. Petersburg Federal Research Center
of the Russian Academy of Sciences (SPC RAS)
St. Petersburg
khar@iias.spb.su

***Abstract*— This paper presents the priority results of interdisciplinary SPC RAS research in the areas of cluster analysis and object detection in a digital image. For a specific domain of video data, the NP-hard problem of estimating *optimal* piecewise constant image approximations, which are characterized by possibly minimal approximation errors (total squared errors) for each number of 1, 2... etc colors, is posed and solved. The novelty of just this paper is in the presentation of accelerating the calculation of optimal image approximations.**

Keywords—image processing, big data cluster analysis, Ward's pixel clustering, Sleator-Tarjan dynamic trees

I. Introduction

Conceptually, we assume that the natural visual system of a person or, say, fly sees the surrounding world optimally and is able to calculate the optimal image approximations in 1, 2, ..., N colors, where N is the number of pixels in the image. Then, to simulate visual perception, the computer should be endowed with this ability. On the other hand, optimal approximations, like pixels, are objective data that depend only on the image and do not depend on generation algorithms, calculation optimization methods, any training or other a priori data about objects. Therefore, in any case, optimal approximations are quite useful for efficient automatic image processing.

For grayscale images, 10 years ago in [1] we successfully solved the problem of generating optimal approximations and continued to solve the problem for color images. Unfortunately, over the past time we have not come across similar studies. Due to the lack of available benchmark samples of optimal color image approximations, we had to obtain them ourselves without using computational speed optimization, which could affect the results of minimizing the approximation error E. At the same time, even for an approximate calculation of optimal approximations in a reasonable time, it was necessary to develop both a meaningful and a computational image model [2-4], because otherwise it is practically impossible to solve an NP-hard problem.

Paradoxically, the novelty of the work lies, first of all, in the fact that when developing object detection programs, we rely on classical methods of cluster analysis [5-8] and classical principles of developing an image segmentation apparatus [9], which is provided by clustering pixels to detect object clusters instead of object instances, similar to "semantic" or "instant" segmentations.

We found that to successfully calculate optimal image approximations, it is necessary to modernize classical clustering methods developed before the advent of computers, as well as take into account the specifics of Big Data that have become available for modern processing.

II. An Image As Polyhierarchical Structure

We use the term *structure* if the image is numerically described by a convex sequence of total squared errors of its approximations in 1, 2, ..., N colors.

The term *polyhierarchical* reflects the specifics of Big Data, which are structured, but are not hierarchical structures and therefore are approximated by the latter ambiguously (Fig. 1).

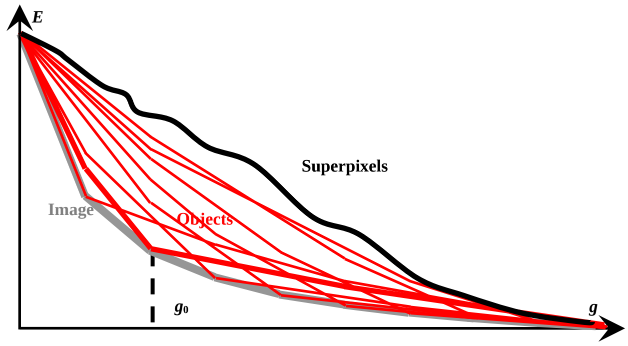

Fig. 1. The multi-valued solution of the problem of image hierarchical approximation, achievable by Ward's pixel clustering. The lower gray convex curve describes E_g sequence of optimal image approximations. The upper non-convex black curve describes errors E_g of image approximations by superpixels constituting some irregular hierarchical sequence. The remaining red convex curves describe E_g sequences of hierarchical image approximations, each containing at least one optimal approximation in a corresponding incrementing number of colors..

Fig. 1 illustrates the dependence of the approximation error on the number of colors in the image. The limiting lower curve describes the errors of optimal image approximations. It is assumed to be convex, which is verified experimentally.

The upper red curves are convex by construction. They intertwine with each other and describe hierarchies of approximations corresponding to objects (each hierarchy of objects contains at least one optimal approximation, and the red curves touch the gray curve at least at one point). The top black sinuous curve describes an unstructured hierarchy of starting elements of image and objects called superpixels [2,3].

It can be shown that such approximation of an image by numerous object hierarchies exists. All that remains is to calculate it.

III. A SYSTEM OF THREE METHODS

The main shortcomings of classical clustering methods in computer processing of Big Data are listed in Table 1.

TABLE I. MAIN SHORTCOMINGS OF METHODS FOR $E=3N\sigma^2$ MINIMISATION

Original Methods	Shortcomings
Ward's pixel clustering	• Instability of the *E*(*g*) resulting values • Excessive computational complexity
K-means	• Unjustifiably rough criterion for *E* minimizing. • The calculation of cluster centers themselves • Reclassification of individual pixels rather than their sets
Split/Merge	Inefficient implementation of reversible computations with an arbitrary hierarchy of pixel clusters

Ward's method is the main one. It is characterized by instability of the approximation errors E_g for each number g of colors from the target range. It is noteworthy that due to multi-iteration calculations, the approximation error E_g changes with slight modification of the image, or a change in the order of scanning cluster pairs that are enlarged differently. It seems that the instability of Ward's method when applied to Big Data remains unnoticed so far [7,8]. Meanwhile, the established instability turns into a very useful property of variability, which allows one to effectively minimize the approximation error E over the entire range of the color number g with repeated application of Ward's clustering, as Monte Carlo method.

The weakest link is K-means method, and among the clustering methods, it is the one that is most intensively used in image processing.

When implementing the simplest split/merge method in an adaptive version, organizational difficulties arise, since this requires an efficient data structure that supports reversible calculations with arbitrary clusters as easily as with individual pixels.

Main disadvantages of conventional data structures describing the hierarchy of clusters are as follows:

- Using dendrograms rather than Sleator-Tarjan dynamic trees [10,11].
- Inefficient implementation of *reversible calculations* with unlimited rollback and reversible merging of pixel sets [12].

The working formula of Ward's pixel clustering for the minimizing increment ΔE_{merge} of approximation error E caused by merging clusters a and b with pixel numbers n_a, n_b and pixel values I_a, I_b averaged within the clusters is as follows:

$$\Delta E_{merge} = \frac{n_a n_b}{n_a + n_b} \| I_a - I_b \|^2 ,$$

where $\| \ \|$ denotes Euclidean distance between three-component pixel values.

Modernization of Ward's pixel clustering comes down to applying this method to pixels subsets for some image partition.

The simplest split/merge method, called CI (Clustering Improvement)-method, uses when splitting cluster the above formula for the increment of the approximation error E caused by merging clusters. But the increment ΔE_{split} is taken with the opposite sign: $\Delta E_{split} = -\Delta E_{merge}$, which implies the reversibility of the cluster merging operation. It is characteristic that CI-method retains the image approximations obtained by Ward's pixel clustering.

The advanced K-means is K-meanless method [13]. S. D. Dvoenko emphasized in the method name that pixel reclassifying should be performed without calculating the intermediate cluster centers. This helps to avoid computing errors such as empty clusters [14].

Working formula for the increment $\Delta E_{correct}$ of the approximation error E, accompanying the reclassification of k pixels with the average I_k value from cluster a into cluster b is expressed as follows:

$$\Delta E_{correct} = k\left(\frac{n_b}{n_b + k} \| I_b - I_k \|^2 - \frac{n_a}{n_a - k} \| I_a - I_k \|^2 \right).$$

From the last expression, the formula for K-means is obtained by eliminating the coefficients that take into account the number of pixels in clusters. At one time it was convenient for calculations using an arithmometer, but not a modern computer. However, when calculating benchmark samples of optimal image approximations it turned out to be possible to do without K-means method altogether thanks to the variability of Ward's pixel clustering.

As for the data structure for generating, storing and transforming hierarchical sequences of image approximations, for such calculations Sleator-Tarjan dynamic trees seem indispensable, since compared to ordinary trees (dendrograms) they have the following advantages:

- Sleator-Tarjan dynamic trees are built on a set of N pixel coordinates without creating additional nodes and therefore occupy half as much computer memory.
- Trees and other metadata are clustered together with structured sets of pixels. At the same time, a network of Sleator-Tarjan dynamic trees, cyclic graphs, pointer systems, additive characteristics and other related data are thrown onto the pixels.
- Tree-structured metadata supports the structuring of images and objects by minimizing approximation errors in forward and backward calculations.

It is important that the three considered modernized methods for minimizing the approximation error E form a system and, in order to achieve maximum speed without loss of clustering quality they are developed for joint use according to certain rules. But at the stage of obtaining benchmark samples of optimal image approximations, the quality of clustering was primary. So, it was achieved by trivially using only Ward's method, and acceleration was ensured through parallel computing.

IV. Experimental Results

The following Fig. 2 and Table 2 reveal the contents of Fig. 1 from the point of view of a software engineer and also explain the output of the primary structuring of the image and objects in the image.

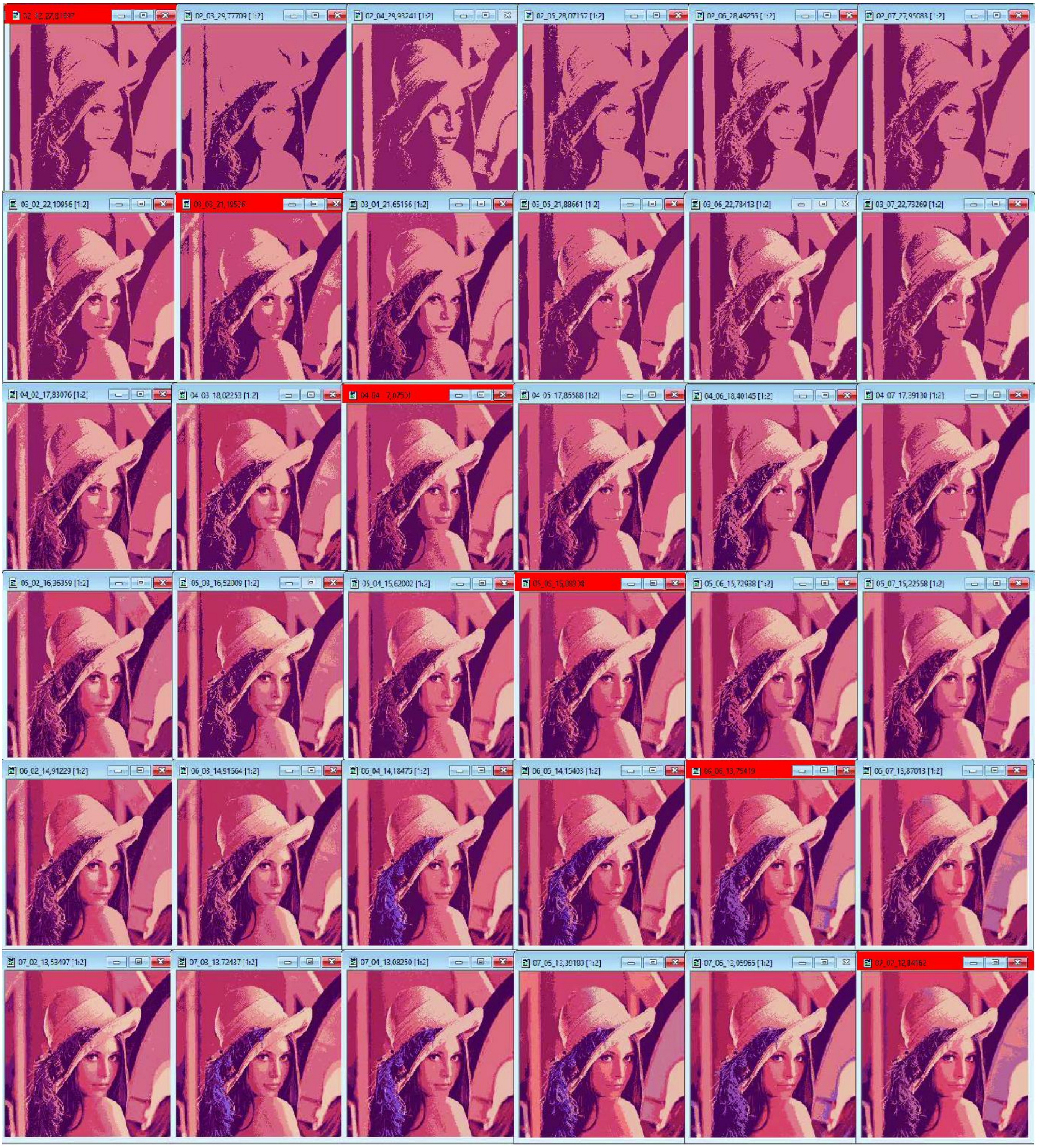

Fig. 2. Dynamic Table of 262144 approximations for "Lena" color image (dimensions 512×512 pixels). The first row and first column of the table were cropped. The columns of Dynamic Table, containing the image approximations in 2–7 colors, are shown. Each column contains a binary hierarchical sequence of image approximations with incrementally added colors: g = 2, 3, 4, 5, 6 and 7. On the main diagonal of the Dynamic Table are the optimal image approximations in g_0= 2, 3, 4, 5, 6 and 7 colors. Image approximations at the top are labeled with g color numbers and corresponding standard deviation σ. The optimal image approximation at the top are marked with red color.

Fig. 2 shows the fragment of so called Dynamic Table that actually illustrates Fig. 1 [2,3]. The calculated hierarchies of image approximations are located in columns of Dynamic Table. When the row number is increased by 1, one of the colors in the current image approximation is split up into two ones. The diagonal approximations are just that are maximally improved in the error E when applying various generation algorithms.

Parameter g_0 is equal to the number of colors in the optimal image approximation and is counted along the diagonal. In the user's view, the entire Dynamic Table of $N \times N$ image approximations is allocated in RAM. In fact, it is encoded in RAM by special data structure and the viewed approximations are generated on-line as needed. That is why the demonstrated table is called a Dynamic Table.

The user's task is to choose a column of approximations in which the structured objects-of-interest are best displayed. According to the user's choice, the tuning parameter g_0 is set up, and the objects are approximated either by unions or by parts of pixel clusters of the optimal image approximation in g_0 colors.

A Dynamic Table contains a sequence of hierarchies of image approximations. Given optimal image approximations, it can be easily generated using Ward's pixel clustering method.

Table 2 explains the same Dynamic Table as Fig. 2, in which the image approximations are replaced by their standard deviations σ.

TABLE II. DYNAMIC TABLE OF N^2 IMAGE APPROXIMATIONS

1	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	46,04981	**46,04981**	**46,04981**	**46,04981**	**46,04981**
2	27,81697	29,77709	29,93241	28,07157	28,49255	27,95083	28,17859	27,90302	29,65169	27,83996	**28,0214**	**29,41112**	**28,45686**	**29,95515**
3	22,10956	21,19506	21,65156	21,88661	22,78413	22,73269	21,80161	22,61297	22,12696	22,29199	**22,42447**	**23,14065**	**23,36193**	**21,99656**
4	17,83076	18,02253	17,07501	17,86588	18,40145	17,3913	17,84489	17,73448	17,45965	17,47786	**18,27078**	**18,06902**	**18,19303**	**17,43498**
5	16,36359	16,52009	15,62002	15,08308	15,72938	15,22558	16,06497	15,73489	16,06194	16,07563	**15,54653**	**15,80576**	**15,69404**	**15,95939**
6	14,91229	14,91664	14,18475	14,15403	13,76419	13,87013	14,53167	14,43118	14,88991	14,56707	**14,26015**	**14,26857**	**13,98714**	**14,52036**
7	13,53497	13,72437	13,0825	13,39189	13,05965	12,94162	13,1255	13,1816	13,63081	13,49235	**13,36412**	**13,52597**	**13,19856**	**13,37849**
8	12,51078	12,46954	12,45969	12,69997	12,53175	12,38413	12,11697	12,40141	12,55824	12,54075	**12,43018**	**12,85797**	**12,65099**	**12,5748**
9	11,75557	11,69727	11,92001	12,2159	11,98996	11,8725	11,40925	11,80964	11,94245	11,92334	**11,75422**	**12,20323**	**12,093**	**12,00841**
10	11,26622	11,15703	11,45123	11,72168	11,56144	11,3584	11,00014	11,23642	11,36592	11,37768	**11,15937**	**11,56803**	**11,53747**	**11,48782**
11	10,85208	10,65161	10,9665	11,2082	11,14244	10,84977	10,68493	10,66212	10,75998	10,81374	**10,76813**	**11,07146**	**11,0615**	**11,08202**
12	10,43215	10,34597	10,57243	10,78406	10,7237	10,46091	10,36246	10,24677	10,28311	10,3209	**10,38843**	**10,5519**	**10,57204**	**10,67417**
13	10,04423	10,07199	10,28153	10,34448	10,29643	10,06842	10,03862	9,94684	9,90688	9,93471	**9,99706**	**10,11099**	**10,15326**	**10,25102**
14	9,70184	9,79683	9,98996	9,94971	9,87209	9,75809	9,72002	9,67179	9,61285	9,60326	**9,60928**	**9,67462**	**9,72393**	**9,86624**
15	9,41357	9,52505	9,69228	9,63151	9,56303	9,45282	9,43502	9,39334	9,33113	9,31958	**9,28883**	**9,34057**	**9,41909**	**9,49333**
16	9,20455	9,273	9,42152	9,30578	9,24672	9,15793	9,15424	9,18172	9,07741	9,09874	**9,03991**	**9,03632**	**9,10572**	**9,16265**
17	8,99841	9,05819	9,15456	9,05354	8,97312	8,86451	8,87031	8,96716	8,89003	8,89583	**8,80141**	**8,79101**	**8,79755**	**8,93524**
18	8,83267	8,84482	8,88235	8,82819	8,69217	8,61745	8,58718	8,76943	8,71926	8,7305	**8,58232**	**8,5934**	**8,54528**	**8,73856**
19	8,67034	8,63201	8,61581	8,63714	8,54538	8,36463	8,3368	8,58661	8,54898	8,5659	**8,38287**	**8,41408**	**8,38127**	**8,55508**
20	8,51405	8,43904	8,36153	8,44724	8,40563	8,17874	8,12188	8,40084	8,37551	8,41027	**8,20242**	**8,24584**	**8,23829**	**8,37974**
21	8,36349	8,25932	8,18764	8,26291	8,27617	8,0119	7,96411	8,23778	8,20258	8,25783	**8,05196**	**8,08257**	**8,10029**	**8,2039**
22	8,221	8,09943	8,0173	8,08737	8,14961	7,86959	7,81877	8,08025	8,03765	8,11387	**7,90863**	**7,93805**	**7,96019**	**8,02635**
23	8,08954	7,94169	7,88748	7,91034	8,02356	7,73866	7,68179	7,94495	7,87533	7,96864	**7,7633**	**7,79897**	**7,82731**	**7,86863**
24	7,96641	7,79921	7,75845	7,75865	7,89673	7,61193	7,56743	7,82181	7,72083	7,83135	**7,63047**	**7,67085**	**7,70786**	**7,737**
25	7,84246	7,66318	7,64224	7,64312	7,77528	7,50645	7,45247	7,69729	7,60714	7,70087	**7,51607**	**7,54784**	**7,59104**	**7,60512**
26	7,71691	7,53469	7,53555	7,52784	7,65352	7,40892	7,36461	7,58215	7,49749	7,58459	**7,40014**	**7,42739**	**7,47678**	**7,47117**
27	7,59836	7,40987	7,43169	7,41952	7,53019	7,31157	7,27725	7,47016	7,39898	7,4856	**7,28891**	**7,3165**	**7,37262**	**7,35019**
28	7,49003	7,31517	7,32764	7,32252	7,41105	7,21744	7,19103	7,36798	7,30591	7,38719	**7,17943**	**7,21739**	**7,27661**	**7,23033**
29	7,38438	7,2211	7,2324	7,22865	7,30895	7,1236	7,11104	7,26464	7,21679	7,30116	**7,07599**	**7,12731**	**7,18168**	**7,13545**
30	7,28007	7,14653	7,1456	7,1365	7,20684	7,04297	7,04123	7,16145	7,12755	7,21744	**6,98297**	**7,04506**	**7,08717**	**7,04373**

TABLE III. DYNAMIC TABLE OF N^2 IMAGE APPROXIMATIONS (CONTINUE)

31	7,19302	7,07657	7,05984	7,04814	7,12264	6,96264	6,97388	7,05924	7,03759	7,13582	6,89512	6,96356	6,99429	6,95862
32	7,10847	7,00829	6,97479	6,9667	7,04217	6,88142	6,907	6,96795	6,95587	7,06302	6,80898	6,88303	6,91448	6,87691
33	7,02365	6,94002	6,899	6,90134	6,96593	6,80094	6,83987	6,88691	6,87833	6,98987	6,72793	6,80168	6,83408	6,79549
34	6,94228	6,87325	6,82272	6,83581	6,89022	6,72944	6,77744	6,82589	6,80768	6,92282	6,65503	6,72502	6,75669	6,71472
35	6,87053	6,80682	6,74967	6,77108	6,81634	6,65926	6,71533	6,76675	6,73633	6,85707	6,58363	6,64897	6,68479	6,63495
36	6,79924	6,74483	6,67829	6,7117	6,74611	6,592	6,65397	6,70871	6,66727	6,79223	6,51382	6,57245	6,61322	6,5554
37	6,72965	6,69246	6,60742	6,65547	6,67903	6,52809	6,59205	6,65046	6,603	6,72741	6,44517	6,50568	6,54538	6,47491
38	6,6639	6,6402	6,54264	6,60043	6,61544	6,46389	6,53067	6,59215	6,54164	6,67445	6,38166	6,44104	6,48236	6,40458
39	6,60081	6,58863	6,47817	6,54679	6,55387	6,40247	6,46949	6,53355	6,48223	6,62331	6,3212	6,37751	6,41924	6,34157
40	6,53853	6,53752	6,4184	6,49335	6,4922	6,34147	6,40944	6,47722	6,42314	6,57308	6,26359	6,31478	6,35753	6,27936
41	6,48171	6,48753	6,35992	6,44284	6,43315	6,28239	6,34934	6,42501	6,37175	6,52423	6,21179	6,25476	6,29846	6,21878
42	6,42442	6,43971	6,30872	6,3948	6,37706	6,22582	6,29054	6,37526	6,32105	6,47631	6,1619	6,20136	6,24052	6,15983
43	6,37503	6,39628	6,25913	6,34641	6,32206	6,17314	6,23382	6,32562	6,27073	6,4285	6,11675	6,15384	6,1893	6,10192

The first column of Dynamic Table, as well as other it's repeating columns, is omitted. An additional column has been added to the left that shows the number of colors for the approximations listed in the corresponding row. The standard deviation values of the optimal image approximations are highlighted in red. It can be verified that the square minimal standard deviations for successive values of the number of colors form a convex sequence. In each row they are the minimum ones.

The computer analyzes the standard deviations of Dynamic Table Fig. 3, while the software engineer analyzes the approximations themselves (Fig. 2) with objects painted in different colors to then use automatic coloring as objects-of- interest attributes.

Fig. 3 explains the testing of the results of accelerating the calculation of optimal approximations, by an order of magnitude of 1000 times, applying Ward's clustering method to "Lena" image parts.

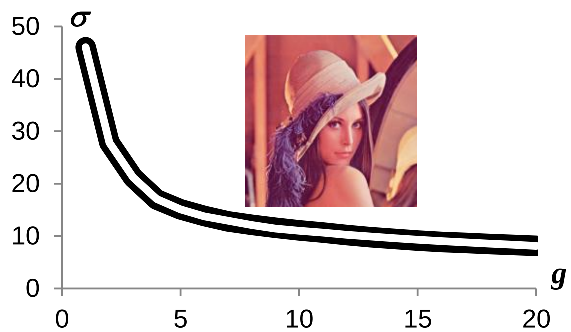

Fig. 3. Verification of Ward's pixel clustering by image parts. The thick black curve describes the control dependence of the standard deviations on the number of colors in the image approximations. The white curve describes a similar dependence for the hierarchy of image approximations obtained by Ward's pixel clustering for N pixels divided into 10 subsets.

The thick black curve shows the dependence of the approximation errors on the number of colors for the control optimal approximations of benchmarks, which were obtained as a result of lengthy calculations using the original Ward's method with various enlargements of the initial pixels and minimizing the approximation errors E_g for the given range of the color numbers g. The white curve shows the dependence of the approximation errors E_g on the number g of colors for the case of dividing the image into 10 parts, which are first processed as independent images, and then the resulting hierarchies of approximations are combined into a single hierarchical sequence of image approximations.

It should be noted that Ward's clustering by parts is principally different from the original Ward's clustering, since when processing an image by parts, pairs of clusters from different parts are ignored. All the more unexpected is the practically coincidence of the obtained sequence of minimal errors of optimal approximations with the reference sequence, as well as the coincidence within the limits of visual indistinguishability of the optimal image approximations themselves with the same color numbers.

V. CONCLUSION

So, in this paper we briefly presented the development of SPC RAS on estimating optimal image approximations for Computer Vision, in particular, we presented pilot results of a high-speed program for Ward's pixel clustering by image parts.

According to [9], the principal requirements for segmentation are:

- The presence of a single algorithm.
- The presence of a numerical criterion for selecting the best of several segmentation options (in our case, segmentation via clustering).
- Scaling invariance.

Regarding the first requirement, it should be noted that, following G. Koepfler, we implement algorithms not as usual, for a obviously limited set of specific images, but for the general domain of images as a whole. We believe that the desired algorithms are quite simple, because accessible to the “intelligence” of a fly.

We strictly adhere to the second requirement.

Experimental verification [15] of the third requirement shows that it is met. This is not surprising, because Ward's method does not take into account the geometric placement of pixels and it is invariant under linear transformation of pixel numbers. Moreover, we have established a tendency for the conversion of an image to grayscale to be commutative with its conversion to an optimal approximation in a given number of colors (Fig. 4).

From a practical point of view it seems useful:

- Modernize classical methods of cluster analysis, as well as their implementation in commonly used software tools such as Matlab for effective computer processing of *Big Data* (polyhierarchical structures).
- To study the objective characteristics of digital images, supplement with optimal approximations at least one of the well-known databases such as Berkeley Segmentation Dataset (BSD).
- Find out which signals, such as audio and other signals have a polyhierarchical structure.
- Adopt the experience of Berkeley University in teaching Sleator-Tarjan dynamic trees.

Optimal color image approximations are available. Let's make them commonly used

Fig. 4. Stability of segmentation through optimal clustering when converting a color image to a grayscale representation (top pair of approximation rows) and invariance to scaling (bottom pair of approximation rows).

ACKNOWLEDGMENT

This research was funded within the framework of the bud-getary theme FFZF-2022-0006 (Theoretical and Technolo-gical Basis for Operational Processing of Big Heterogeneous Data Streams in Sociocyberphysical Systems).

REFERENCES

[1] M. V. Kharinov “Image segmentation method by merging and correction of sets of pixels. Pat. Recog. Image Anal. Adv. Math.Theory Appl. 2013, 23, 393–401.

[2] I. G. Khanykov, V. A. Nenashev, M. V. Kharinov “Algebraic Multi-Layer Network: Key Concepts. ” J. Imaging 2023. 9, 146

[3] V. A. Nenashev, I. G. Khanykov, M. V. Kharinov “A Model of Pixel and Superpixel Clustering for Object Detection.” J. Imaging 2022, 8, 274.

[4] V. A. Nenashev, I. G. Khanykov “Formation of fused images of the land surface from radar and optical images in spatially distributed on-board operational monitoring systems. ” J. Imaging, 2021 7(21).

[5] S. A. Aivazian, V. M. Bukhshtaber, I. S. Eniukov, L. D. Meshalkin, “Prikladnaia statistika: klassifikatsiia i snizhenie razmernosti [Applied statistics: classification and dimension reduction].” Finance and Statistics Publisher: Moscow, Russia, 1989. p. 607.

[6] I. D. Mandel "Klasternyi analiz [Cluster analysis]." Finance and Statistics Publisher: Moscow, Russia, 1988. p. 176.

[7] F. Murtagh, P. Legendre "Ward's hierarchical clustering method: clustering criterion and agglomerative algorithm." arXiv 2011, arXiv:1111.6285.

[8] A. Fernández, S. Gómez "Versatile linkage: a family of space-conserving strategies for agglomerative hierarchical clustering. " J. Classif. 2019, 37, 584–597.

[9] G. Koepfler "Segmentation by minimizing functionals and the merging methods," in World Congress of Nonlinear Analysts'92, Proc. of the First World Congress of Nonlinear Analysts, Tampa, FL, USA, 19–26 August 1992; Lakshmikantham, V., Ed.; DeGruyter: Berlin, Germany; Boston, MA, USA, 1996; pp. 1933–1942.

[10] D. D. Sleator, R. E. Tarjan "Self–Adjusting Binary Search Trees." Journal of the ACM, 1985. 32(3), 652–686.

[11] R. Nock, F. Nielsen "Statistical Region Merging." IEEE Trans. Pattern Anal. Mach. Intell. 2004. 26(11), 1452–1458.

[12] T. Toffoli "Reversible computing," in International Colloquium on Automata, Languages, and Programming; Springer: Berlin/Heidelberg, Germany, 1980; 632–644.

[13] S. D. Dvoenko "Meanless k-means as k-meanless clustering with the bi-partial approach," in 12th International Conference on Pattern Recognition and Image Processing (PRIP'2014), UIIP NASB, Minsk, Belarus, 24–27 September 2014; pp. 50–54.

[14] D. Aloise, N. C.Damasceno, N. Mladenoviĉ,; D. N. Pinheiro "On strategies to fix degenerate k-means solutions." J. Classif. 2017, 34, 165–190.

[15] M. V. Kharinov "Polyhierarchical image structure and invariant object detection," in Graphicon-Conference on Computer Graphics and Vision, Moscow, 2023 (in press)