computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

IMAGE STRUCTURE ANALYSIS BY 3-STAGES CLUSTERING

Roman Melnyk [1)], Ruslan Tushnytskyy [2)]

[1)] Professor, Software Department, Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013,
ramelnyk@polynet.lviv.ua
[2)] Post-graduate student, Software Department, Lviv Polytechnic National University, 12, S. Bandery str., Lviv, 79013,
ruslan.tushnytskyy@gmail.com

Resume: *An approach for decomposition of visual images by clustering and breaking them down into geometric figures is considered. Multilevel hierarchical clustering algorithm to form three emphasized levels of clusters such as rectangles, closed regions and integrated areas is proposed. Advantages of such decomposition in three stages are as follows: images covered by rectangles are planned to be formatted and compressed, image fragments could be taken for the preliminary pattern recognition or could easily be corrected, hierarchically constructed fragments are good material to form pattern features for searching procedures. The algorithm complexity, the proposed approach of scanning searching area to reduce it, the rolling up criteria and key parameters for its control are investigated. The results of pattern analysis by structure features for some practical problems are presented in the article***.**

Keywords: *cluster, clustering, visual pattern, hierarchical tree, rolling-up algorithm, scanning area, rectangles, integrated areas, structured images, structural features.*

1. INTRODUCTION

The clustering methods are widespread for visual patterns segmentation, data mining, indexing and searching. [1−9]. In particular, the work [1] contains the classification of the clustering methods and the approach for contouring chosen clusters. The works [2], [3] describe graph's models for images and clustering algorithms for its segmentation. The works [4], [5] are devoted to the clustering approach for experimental data covered by a grid with predetermined step. Hierarchical methods were used for decomposition and enhancement of medical images and fingerprints [6] − [7]. Various approaches are used for improving and decomposition of images, among them being assessment of image fragment local characteristics [8], forming regions of special figures [6] or clustering according to criteria of statistical dependencies [7]. These works, as well as a number of others, do not consider the variety of control parameters, by which researcher could investigate different dependences of clustering results, e.g., such as brightness threshold and intensity sensitivity for number of clusters to be formed. As a rule they are based on one stage clustering procedures.

Different agglomerative methods differ by the way that similarity between two clusters is updated. If the linkage satisfies a cluster aggregative inequality then the algorithm can be implemented efficiently at a time complexity of $O(N^2)$ [10]. A new recent method [11] uses a sophisticated merging criterion which takes the cluster internal structure into account. The multilevel hierarchical clustering algorithm in [12] are mainly planned for data analysis, has one stage button-up procedure and proposes new linkage criteria to reduce the computation complexity from $O(N^3)$ to $O(N^2)$.

Standard agglomerative clustering methods use ordinary similarity distances between clusters to be merged. The proposed approach is based on a number of merging criterion: value of common border between clusters, difference between their weighted brightness, difference between square values, etc. Moreover, merging is being performed if constraints allow it. The work is based on three stages clustering algorithm: covering an image by rectangles, closed regions and integrated areas Constraints are formed to control form of rectangles or brightness of closed regions and integrated areas.

Some investigations were held to describe different dependences: number of clusters, parameters of closed regions and integrated areas due to criteria for merging: cluster brightness threshold, algorithm sensitivity or shape and dimension constraints.

The full procedure allows to formalize visual image description by breaking them down into integrated areas, closed regions or rectangles [Fig.1].

An access to output fragments of every stage allows them to be corrected and broken into fragments. We would like to notice that we didn't find such image fragments classification in the referenced papers and others read. Additionally we note that *Microsoft Visual Studio.NET* disposes a class Region to paint figures with non regular shape.

When applying three stages clustering algorithm to practical images algorithmic complexity and computation time of image decomposition process would be taken into account. For this reason, scanning searching procedures on different stages of the algorithm are developed.

For large database with over tens of thousands of visual patterns, effective indexing becomes an important issue the content-based image retrieval systems (CBIR). The existent universal CBIR systems attribute to one of three categories depending on approach of extracting features, for example color [15 − 16], shape [17 − 18], structure and location [19 − 20]. Presented 3-stages clustering algorithm extracts some pattern structural properties (pattern features) which were taken to analyze and classify images.

2. TASK OF IMAGE DECOMPOSITION

The tasks of the decomposition algorithm when being applied to visual patterns are to solve the following problems: 1) to get full description for all parts of image with its all possible characteristics; 2) to ensure an access for every part of pattern independently of its position on three levels hierarchical tree: integrated areas → closed regions → rectangles. Such access is planned to be used in searching procedures for pattern and its fragment recognition.

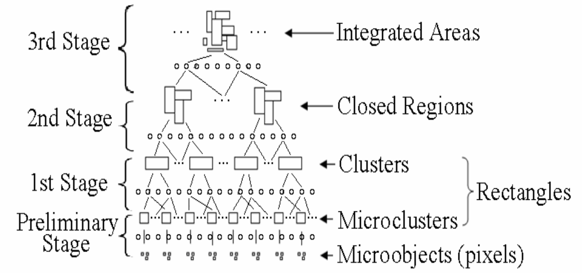


Fig. 1 – Fragment of multilevel hierarchical tree

To solve these complex tasks the multilevel three stages clustering algorithm based on traditional binary hierarchical tree is developed. Three stages of the process, the nodes of interest and a small fragment of the multilevel tree are shown on Fig. 1. Three stages are emphasized because they have different searching criteria to build different geometric figures. For user it is more interesting to consider three levels tree extracted from full tree presented on Fig. 1 with earlier defined pattern fragments (areas, regions and rectangles) as nodes.

Before the clustering algorithm begins its work the preliminary procedure to prepare leafs of the hierarchical tree has to start. Non divisible microobjects, i.e., pixels of digital image must be united to form microclusters − smallest non divisible rectangles. The microclusters are fully or partly filled [Fig. 2(a)] in the case of black-and-white image or are combined from pixels of different brightness for grey image [Fig. 2 (b)].

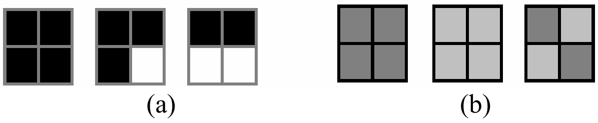


Fig.2 – Black-and-white and grey clusters

On the preliminary stage we must to solve the problem:

It is necessary to obtain a partitioning $\bar{O}$ of the set of microobjects $O=\{o_1, o_2, \dots , o_N\}$:

$$\left.\begin{aligned} &\overline{O} = \{O_1, O_2, ..., O_n\}, \\ &n \to \min, \end{aligned}\right\} \qquad (1)$$

so that a total number of partitions is minimized while satisfying the constraints:

$$F_k^- \le F_k(X_i) \le F_k^+, \quad (k = \overline{1,m}; i = \overline{1,n}), \qquad (2)$$

where n is number of microclusters O_1, O_2, … , O_n; F_k^- , F_k^+ − are boundary values of the k-th feature function. The functions F_1, F_2, …, F_n define coordinates, brightness, dimension, number of pixels, shape, color, ratio between filled and empty cells etc.

The cluster features or parts of them are also assigned for clusters on the higher levels of the hierarchical tree. So to make pattern decomposition by rectangles, closed regions and integrated areas we must to solve the problem (1) three times. We solve the optimization problem by the rolling-up procedure, i.e., algorithm of bottom-up merging of clusters to get minimal number of segments covering a pattern. The stages differ between themselves by merging criterion and constraints. The rolling-up procedure divides a set of objects into groups having the same or similar characteristics. Tree nodes not satisfying the constraints and subsequently not being the subjects for further rolling-up process are transferred to the result set of clusters.

The visual pattern area is being covered by grid with steps selected from a set of values: 1×1, 1×2, 2×1, 2×2 etc. Some scanning procedures to solve the problem (1) on the preliminary stage of the algorithm and to build microclusters as rectangles sized by selected steps of grid are developed. A cluster number on higher stages depends from a microcluster number and its features. That is why we developed alternative algorithms for scanning procedures on the preliminary stage of the algorithm. The trivial approach to get microclusters is to describe a grid cell as microcluster, i.e., to define for it all features and to assign them to it. For reason of a lack of paper place other preliminary scanning algorithms are not considered in the paper.

3. CONTROL PARAMETERS

For grey image every microobject (pixel), is characterized by brightness value ranging from black to white. Brightness b of microobject we define by equation expressing its relative filling as percentage of black color:

$$b = (256 - c_i) \times 100 / 256, \qquad (3)$$

or as percentage of white color:

$$b = c_i \times 100 / 256 \qquad (4)$$

where c_i (i = 1,2,3) is a value of one from the R (G or B) components of grey pixel.

Fig. 2(b) shows rectangles by different brightness values.

The equations (3), (4) are applicable only for microobjects. To denote integral microcluster brightness we use $B_{rk.}$ We define it as a mean weighted value for two microobjects (microclusters) indexed by i and j and merging to a microcluster numbered by k:

$$B_{rk} = \frac{b_i \times a_{i1} \times a_{i2} + b_j \times a_{j1} \times a_{j2}}{a_{i1} \times a_{i2} + a_{j1} \times a_{j2}}, \qquad (5)$$

where b_i, b_j are the brightness values, a_{i1}, a_{i2} a_{j1}, a_{j2} are values of microobject geometric sizes.

The equations (1) – (5) are being used on the preliminary and 1st stages of the algorithm where microcluster and cluster as rectangles are considered. Fragment of medical pattern clustered by rectangles is illustrated by Fig. 3.

To denote integral closed region and integrated area brightness we use B_{Rk} and B_{Ak}. We define them as a mean value of cluster or closed region brightness of all n objects creating closed region or integrated area:

$$B_{Rk} = 1/n \times \sum b_i; \qquad B_{Ak} = 1/n \times \sum B_{R\,i} \qquad (6)$$

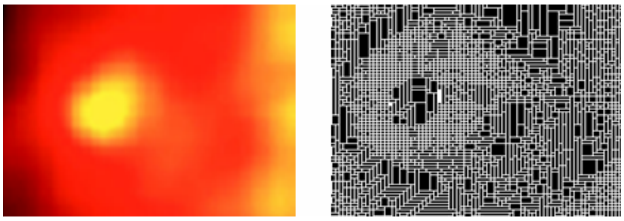


Fig.3 – Clustered pattern. (a) Original. (b) Pattern after preliminary and 1st stages of the algorithm

The equations (6) are being used on the 2nd and 3rd stages of the algorithm where closed regions and integrated areas are considered. Fragment of medical pattern clustered by regions and areas are illustrated by Fig. 4.

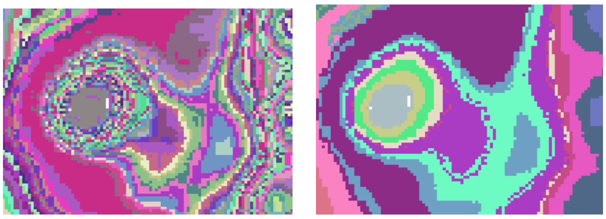


Fig.4 – Clustered pattern after 2nd (a) and 3rd (b) stages of the algorithm (colors are random)

4. ALGORITHM ACCURACY

For black-and-white image there are two types of microcluster (rectangle): completely black and with a number of white positions.

Ratio between a number of black pixels n_{bi} entering i^{th} microcluster and its total number of pixels n_{ti} we use as i^{th} cluster brightness: $r_{bi} = n_{bi} / n_{ti}$. For example, Fig. 2(a) shows for microcluster brightness values 1, ¾, ½.

We note: S_P – area of pattern (number of black pixels), S_{oi} – area of cluster, S_{Pc} – area of the pattern got as coverage by rectangles: $S_{Pc}=\sum S_{oi}$, where i runs over full set of clusters.

The brightness parameter of rectangle we use to control accurate or approximation clustering results. For accurate clustering we have $S_{Pc} = S_P$.

In practice we have to reduce extra or noise information and then it is useful to consider the case $S_{Pc} < S_P$. We call this procedure cleaning or reduction. This process is similar to filtration of thin effects in visual patterns.

Opposite case is if $S_{Pc} > S_P$. This procedure is named as renewal or addition. We mean here, that by a noise in the pattern some needed pixels are destroyed or lost and its picture does not correspond to generally accepted one.

The brightness parameter is a key to control ratio between S_{Pc} and S_P, i.e., to control: 1) a numbers of microclusters and clusters not taking part in merging of the rolling-up process; 2) a number of white pixels included in the clusters covering the pattern; 3) numbers of clusters and closed regions got by pattern clustering.

We consider accuracy for image covered by clusters or covered by integrated areas:

$$E_C = S_{Pc} / S_P, \qquad E_{IA} = IA_b / IA_{100}. \qquad (7)$$

where IA_b – number of integrated areas, got by clustering with user defined brightness parameter b, IA_{100} – number of integrated areas, got by clustering with brightness parameter b = 100%.

5. DECOMPOSITION ALGORITHM

To solve the problem (1) we consider the rolling-up algorithm having input objects as microclusters, clusters and closed regions and output objects as clusters, closed regions and integrated areas. To use one description for all these object we call them as Clusters. We denote them by $Q_1, Q_2, \ldots$. The set X_i belongs to the i-th level of a tree and contains Clusters. On the *i-th* step of the algorithm merging Clusters create new Clusters by which the set X_{i+1} is being filled.

The sequence of steps in the ***rolling-up algorithm*** is as follows:

0. Prepare initial data: i=1, the set X_i is being filled by input Clusters; $C_i = X_i$, $B_i = \emptyset$.

1. Form for every $Q_k \in X_i$ a set of touching neighbors within the grid by searching procedure.

2. Calculate the criterion function values for all pairs of adjacent Clusters.

3. For every two candidates to be merged and satisfying all constraints create the list $L(F_{ki})$ of criterion values by decreasing order.

3. Delete from $L(F_{ki})$ pairs having intersected indexes beginning to search them from bottom.

4. Merge first pair of Clusters from the list and put new Cluster to the set B_i, correct the set C_i, form the new set X_{i+1} of Clusters.

5. Increasing i=i+1 repeat steps 1–4 until merging takes place.

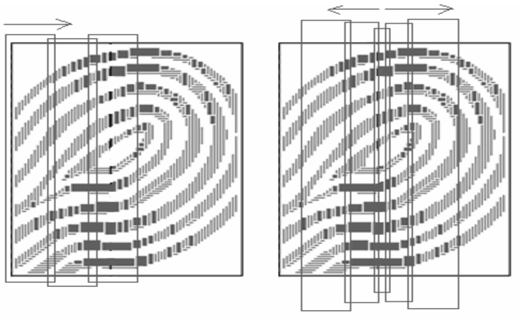
The last set X_L contains output Clusters

6.ALGORITHM COMPLEXITY OPTIMIZATION

Computation complexity of the clustering algorithm could be determined by a number of criterion function calculation, comparison and merging number. In the worst case, when merging number is 2N-1 (N is full number of input Clusters) complexity of clustering algorithm exceeds $O(N^2)$. Theoretically to calculate all functions (needed and extra) it does cost $O(N^3)$. The main operations of the algorithm are related with searching of candidates for merging.

To reduce the algorithm complexity we could mainly at the stage of searching Cluster candidates to be merged. The searching area for the rolling-up procedure we plan as a part of full pattern space. This part is as a rectangle moving through the pattern rectangle by alternative rules. It scans the whole surface of image [see Fig. 7]. Area limits for searching Clusters – candidates to be merged – are put on one axe of coordinates. Decreasing input data volume we reduce the algorithm complexity from value $O(N^2)$ to a set of smaller values $O(n_i^2)$, because the number n_i of Clusters in scanning area is smaller than the number N in full pattern area, i.e., n_i<N. We would like to mention that searching procedure repeats its work step by step covering the whole pattern area by a sequence of smaller scanning areas.

All Clusters of two coordinates pattern space from Fig. 7 are stored in linear data structure containing description of the Cluster model and a pointer for neighboring Cluster [Fig. 8].



(a) (b)

Fig.5 – Scanning area of the algorithm

First rule for scanning is illustrated by Fig. 5(a). The dashed-lined rectangle defines the scanning working area restricted by the right border. Scanning to search for neighboring Clusters is from left to right within the rectangle. After some searching and merging steps the rectangle is being moved to right.

Fig. 5(b) illustrates the second rule of scanning. The rectangle is being restricted both by the right and the left borders. An image is being scanned from left to right within the rectangle. To search for neighboring Clusters the rectangle itself has two directions for moving: to left and to right. Cluster for which neighboring Cluster are being searched we accept as the center of the scanning rectangle.

The scanning area has adaptive property: dimension of scanning rectangle could be of image size or could be formed dynamically depending from Cluster size. For example, at the beginning the scanning area size is being initialized by layout grid steps and further they are being increased step by step. This adaptive property essentially reduces for current Cluster a number of neighbors to be compared.

The experiments presented in the Table 1 confirmed efficiency of scanning area: minimal time economy is by one order and computation advantage is better for bigger images.

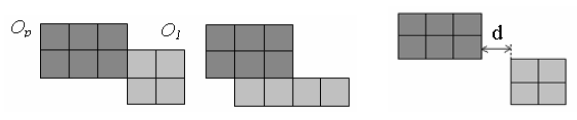
Table 1. Clustering algorithm without and with scanning area

Image dimension	Cluster number	Time without scanning area, s	Time with scanning area, s
120x178	5 155	0,751	0,07
149x150	7 311	0,661	0,05
175x216	8 210	0,450	0,05
141x169	11 952	1,112	0,07
156x173	13 617	2,193	0,09
171x213	14 136	3,074	0,140

7. CLUSTERING TO CLOSED REGION AND INTEGRATED AREAS

By the 2nd and 3rd stages of the algorithm we solve the problem (1) having rectangles $Q(Q_1, Q_2, Q_3, \ldots, Q_N)$ and closed region*s* $R(R_1, R_2, R_3, \ldots, R_N)$ instead of set O. A task for the rolling-up procedure is to find partitioning $\check{R}=\{R_1, R_2, \ldots, R_n\}$ and $\check{A}=\{A_1, A_2, \ldots, A_n\}$ instead of set $\bar{O}=\{O_1, O_2, \ldots, O_n\}$ so that a total number of partitions is minimized: $n\rightarrow\min$, while satisfying the constraints from (1).

Every set R_i contains rectangles included to one closed region by the rolling-up procedure. Criterion for rectangles to be merged is a number of common edges [Fig. 6(a)].



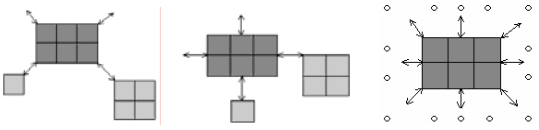
(a) (b)

Fig.6 – Merging criterion (a) and distance between clusters (b)

Constrains are formulated by feature functions $F_{i..}$ Second criterion for Cluster rectangle to participate in merging is minimal distance d in pixels between their closest rectangles [Fig. 6(b)]. Distance d is measure to be controlled by user.

We extended the distance criterion for "diagonal" and "main" distances [Fig. 7(a)].

In particular, diagonal distance is useful to process images having ring structure, e.g., blood cells, fingerprints etc.



(a) (b)

Fig.7 – Diagonal and main distances between clusters (a), a set of coordinates for a neighboring Cluster (b)

To realize the algorithm to solve the problem (1) with such objects as rectangles and closed regions we use new procedure to search candidates to be merged. The procedure is being used on the 1st step of the ***rolling-up algorithm***. The main its steps are as follows:

1. Select a current primary Cluster not belonging to any result Cluster.
2. Form a set of coordinates P around the Cluster from 1st step. The set includes all possible points, where neighboring Clusters could be located including the diagonal and main distances [Fig. 7(b)].
3. Look for a secondary Cluster to select it while satisfying the following conditions:

– it does not belong to any result Cluster;

– one or more coordinates of set P belong to the secondary Cluster.

4. Repeat step 3 until neighboring secondary Cluster exists.
5. Repeat all step until primary Cluster exists.

Having the candidates list the ***rolling-up algorithm*** continues its steps to build results Clusters (regions and areas) by hierarchical tree.

8. STRUCTURE FEATURES

As results the 3-stages clustering algorithm extracts from image some characteristics which we assign to the set of structural features. In particular: MC − number of microclusters, C, CR, IA − numbers of clusters, regions and areas from three stages. Also we get brightness parameters and square values for all objects.

We use these numbers and parameters to define structural properties by which one could estimate the structure degree of image, in particular:

1) structure coefficients corresponding to merging degree by one stage of the algorithm (an index specifies the algorithm stage):

$$K_s^1 = C / MC, \quad K_s^2 = CR / C, \quad K_s^3 = IA / CR \quad (8)$$

2) structure coefficients corresponding to merging degree by two stages of the algorithm:

$$K_s^{12} = CR / MC, \qquad K_s^{23} = IA / C. \qquad (9)$$

3) structure coefficients corresponding to

merging degree by three stages of the algorithm:

$$K_s^{123}(MC) = IA / MC. \qquad (10)$$

To characterize inverse property (fuzziness) we use inverse values for (2-5), e.g.:

$$K_f^{123}(MC) = MC / IA. \qquad (11)$$

The image structure coefficients we also assign to structural features. In some cases to widen the set we can in formulas (4-6) to use a total number of image pixels *PX* instead of *MC*:

$$K_s^{123}(PX) = IA / PX. \qquad (12)$$

Each selected fragment i, in particular, area or region characterize the relative number of pixels:

$$vx_i(CR) = px_i(CR)/PX, \ \ vx_i(IA) = px_i(IA)/PX, \quad (13)$$

for which inequality is as follows:

$$\sum px_i(CR) \le PX, \quad \sum px_i(IA) \le PX, \qquad (14)$$

where sum takes for all formed areas IA or region CR.

Some statistical properties could easily be considered as structural properties. For example:

1) Mean cluster, region or area dimension :

$$M(CR) = (1/CR) \cdot \sum px_i(CR),$$
$$M(IA) = (1/IA) \cdot \sum px_i(IA), \qquad (15)$$

2) Cluster, region or area deviation

$$D(CR) = \sqrt{(1/CR)\cdot\sum(px_i(CR) - M(CR))^2},$$
$$D(IA) = \sqrt{(1/IA)\cdot\sum(px_i(IA) - M(IA))^2} \qquad (16)$$

3) To estimate the image elementary object size/deviation features on different levels of decomposition we propose statistical relations:

$$R(CR) = \frac{M(CR)}{D(CR)}, \quad R(IA) = \frac{M(IA)}{D(IA)} \qquad (17)$$

And for full image integral statistical relations

$$R_I(CR) = \frac{IA \cdot M(CR)}{D(CR)}, \quad R_I(IA) = \frac{IA \cdot M(IA)}{D(IA)} \qquad (18)$$

All or part of above mentioned structural features are planned to be used for pattern and its fragments analysis and classification.

9. IMAGE STRUCTURE AND SURFACE ANALYSIS

The clustering algorithm was applied for material structure and surface analysis by using proposed structure features. Work [21] presents structured and non structured images investigation by the grid step and microcluster brightness parameter. Fig. 8 presents samples with lines by different width and length to imitate changing of structure complexity. Table 2 contains some structure coefficients and features illustrating structural changes especially by values of $K_s^{123}(MC)$ and $R_I(IA)$.

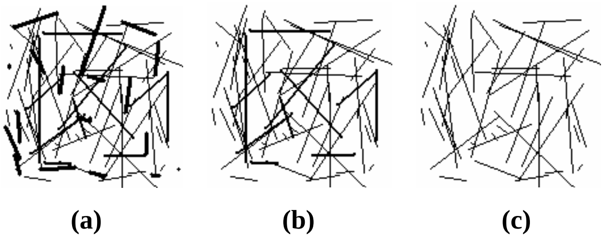


Fig. 8 – Images for structure analysis

Table 2. Structure coefficients and features

Structural characteristics	Visual pattern		
	a	b	c
$K_s^{123}(MC)$	0,00102	0,00214	0,00425
$R_I(IA)$	2,42940	2,87323	3,77030
MC	3912	2802	1882
C	1038	890	707
CR	11	11	13
IA	4	6	8

Fig.9 presents medical pattern structure analysis. The determined structural features of the mattern could be used for its indexing ans searchin in databases as well as for qualitative and quantitative estimation of the pattern given.

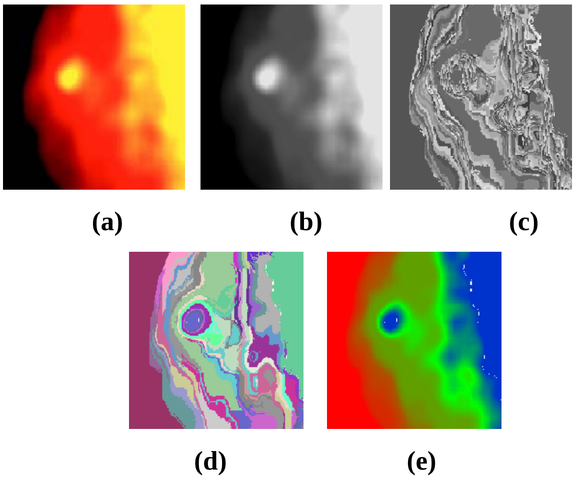
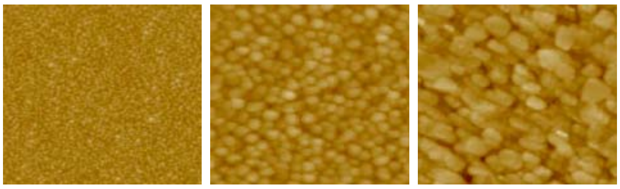


Fig.9 – Medical pattern. (a) Input image. (b) Clusters. (c) Closed regions. (d) Integrated areas. (e) Closed regions coloured by their weighted brightness

The main structural coefficients and features are as follows: $K_s^{123}(MC)$ = 0,00182, MC = 36241, C = 25384, CR = 25272, IA = 66, $M(IA)$ = 549,106, $D(IA)$ = 1330,06.

By the experiment we can conclude that numbers of final Clusters (integrated areas) are of close values as it was expected.

The algorithm was applied for surface topology of freshly deposited surfaces [Fig.10].



(a) (b) (c)

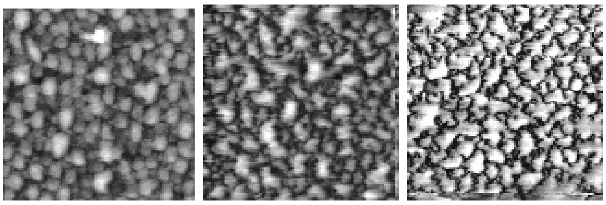
Fig. 10 – Surface topology of freshly deposited (a) Pt, (b) Au and (c) Ag surfaces. The average grain size is 3 nm for Pt, 8 nm for Au, 15 nm for Ag

By structural features from table 3 we can estimate the surface structure changes caused by different grain size, applied to polish the surface.

Table 3. Structure characteristics of metal surface

Structural characteristics	Visual pattern		
	a	b	c
$K_s^{123}(MC)$	0,00244	0,00294	0,00373
$R_I(IA)$	72,641	78,92	99,250
MC	21700	21700	21700
C	17498	15699	15449
CR	13973	11523	10895
IA	53	64	81
$M(IA)$	409,433	339,062	267,901
$D(IA)$	298,726	274,952	218,638

Close results were got by investigating the glass surfaces having different sizes of nanobubles [Fig.11]. Table 4 confirms: the more structure complexity the greater values have structural features.



(a) (b) (c)

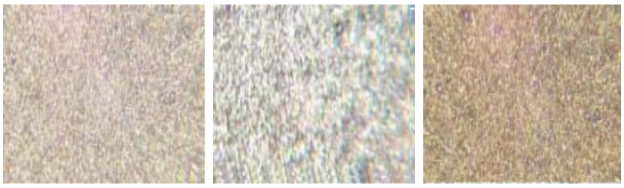
Fig. 11 – A series images revealing nanobubbles on a glass surface

Fig 11(a) correspond to pH 9.4, Fig. 11(b) – pH 4.4, Fig. 11(c) – pH 5.6.

To widen a range of the algorithm application the specimen of polished Si surfaces were taken for investigation (fig12). Samples were elaborated by different grain sizes.

Table 4. Structure characteristics for glass surface

Structural characteristics	Visual pattern		
	a	b	c
$K_s^{123}(MC)$	0,00661	0,00754	0,00870
$R_I(IA)$	103,87	106,225	129,429
MC	11183	11134	9770
C	11112	10692	8836
CR	11072	10463	8076
IA	74	84	85
$M(IA)$	151,121	132,547	114,941
$D(IA)$	107,653	104,814	75,484



(a) (b) (c)

Fig. 12 – Surface images of a Si specimen following lapping with various abrasive types and sizes: (a) 14 – μm BC; (b) – 8 μm BC; (c) – 8 μm SiC

Results in table 5 and others which could be calculated confirmed that they are applicable for quality estimation of polished surfaces.

Table 5. Structure characteristics for Si surface

Structural characteristics	Visual pattern		
	a	b	c
$K_s^{123}(MC)$	0,00307	0,00388	0,00400
$R_I(IA)$	82,589	98,216	89,944
MC	22119	20079	22243
C	18834	17213	19226
CR	16112	14909	16746
IA	68	78	89
$M(IA)$	325,279	257,423	249,921
$D(IA)$	267,818	204,436	247,298

10. PROGRAM PACKAGE

The experimental program package with the user interface has been developed [21]. It controls all operation stages: image input, parameters, intermediate and output images, clustering options (full, specified), report of the algorithm work, result parameters for each clustering stage.

The program packages is programmed in C# language of *Microsoft Visual Studio 2008*.

11. CONCLUSION

The clustering algorithm for visual patterns analysis is developed and investigated. It is based on ideas: 1) classifications of pattern fragments as microclusters, clusters, closed regions and integrated areas; 2) rolling-up algorithm to solve the problem of optimal coverage on three stages of the algorithm; 3) scanning area approach to reduce the algorithm complexity; 4) calculation of structure features to analyze material structure and surface.

The test results confirmed possibility to apply the program package for image analysis and processing to determine their properties and features, to find structural changes, to search defects in objects and materials. The package could be useful for specialists from medicine, metallurgy, spectroscopy and many others making conclusion by sample images.

All tests were performed by a computer with processor *AMD Athlon* 1.68 GHz, 768 Mb RAM.

12. REFERENCES

[1] A. Yip, C. Ding, T. Chan. Dynamic Cluster Formation Using Level Set Methods, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28 (6) (2006). p. 877−889.

[2] L. Grady, E. Schwartz. Isoperimetric Graph partitioning for Image segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28 (3) (2006). p. 469−475.

[3] M. Pavan, M. Pelillo. Dominant sets and Pairwise Clustering, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(1) (2007). p. 167−172.

[4] Z. Yu, H.-S. Wong. GCA: A real-time grid based clustering algorithms for large data set, *Proc. of the 18th International Conference on Pattern Recognition (ICPR)* (2006). p. 740−743.

[5] K. Wilamowska, M. Manic. Unsupervised pattern clustering for data mining, *Proc. of the 27th Annual Conference of the IEEE Industrial Electronics Society (IECON)* (2001). p. 862–1867.

[6] S. Katz, A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts, *ACM Transactions on Graphics* 22 (3) (2003). p. 954–961.

[7] R. Dosil, X.M. Pardo, X.R. Fdez-Vidal. Decomposition of three-dimensional medical images into visual patterns, *IEEE transactions on biomedical engineering* 52 (12) (2005). p. 2115−2121.

[8] L. Hong, Y. Wan, A. Jain. Fingerprint image enhancement: Algorithm and performance evaluation, *IEEE Transactions on Pattern Analysis and Machine Intellegence* 20 (8) (1998). p. 777−789.

[9] K. Anil, R. Dubes. Algorithms for Clustering Data, *Prentice Hall*, 1988. 320 p.

[10] G. Karypis, E. Han, V. Kumar. Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling, *Computer* 32 (1999). p. 68−75.

[11] C. Ding, X. He. Cluster Aggregate Inequality and Multilevel Hierarchical Clustering, *Proc. 9th European Conf. Principles of Data Mining and Knowledge Discovery* (2005). p. 71−83.

[12] M. Laszlo, S. Mukherjee. A Gegridic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-means Clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (4) (2006). p. 533−543.

[13] A. Vailaya, A.K. Jain, H.J. Zhang, “On image classification: city vs. landscape”, *Pattern Recognition*, vol. 31, p. 1921-1935, 1998.

[14] M.J. Swain,D.H. Ballard, “Color indexing”, *International journal of Computer Vision*, vol. 7, n. 1, p. 11-32, 1991.

[15] H. Nezamabadi-pour, E. Kabir, “Image retrieval using histograms of unicolor and bicolor blocas and direccional changes in intensity gradient”, *Pattern Recognition Letters*, vol. 25, n. 14, p. 1547-1557, 2004.

[16] F. Mokhtarian, S. Abbasi, “Shape similatity retrieval under affine transforms”, *Pattern Recognition*, vol. 35, p. 31-41, 2002.

[17] A.K. Jain, A. Vailaya, “Image retrieval using color and shape”, *Pattern Recognition*, vol. 29, n. 8, p. 1233-1244, 1996.

[18] B.S. Manjunath, W.Y. Ma, “Texture feature for browsing and retrieval of image data”, *IEEE PAMI*, vol. 8, n. 18, p. 837-842, 1996.

[19] J.R. Smith, C.S. Li, “Image classification and quering using composite region templates”, *Academic Press, Computer Vision and Understanding*, vol. 75, p. 165-174, 1999.

[20] J.Z. Wang, J. Li, G. Wiederhold, “SIMPLIcity: semantic sensitive integrated matching for picture libraries”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, n. 9, p. 947-963, 2001.

[21] R.A. Melnyk, R.B. Tushnytskyy, “Pattern Analysis by Clystering”, *Proc. of the 5th Intern. Conf. Neural Networks and Artificial Intelligence (ICNNAI)*(2008). p. 160–163.

Roman Melnyk *Professor of Software Engineering Department, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University.*

Research interests: combinatory problems, methods of fuzzy clustering, coding and recognition, storage and data mining.

***Ruslan Tushnytskyy** – a post-graduate student at the Software Engineering Department, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University.*

Research interests: preliminary processing of images, coding and recognition in real time, clustering algorithms, data compression.