11

Block-Diagonal Forms of Distance Matrices for Partition Based Image Retrieval

Dmitry Kinoshenko, Vladimir Mashtalir and Elena Yegorova
Kharkov National University of Radio Electronics
Ukraine

1. Introduction

Rapid expanding of the multimedia data collections volume brings forward the need for visual data efficient organization, storing and search methods. It stipulated diversity of investigations directed to creating efficient image retrieval methods satisfying performance speed and validity requirements. The indexing efficiency is generally evaluated by parameters of storage access number, computational expenses of the index structure search and number of operations for the distance computation between query representation and objects in the database. As image processing requires a lot of time and resources the most efficient practical way to reduce search time expenses is creation of indexing structure on the preliminary processing stage. Unfortunately, existing indexing methods are not applicable to a wide range of problem-oriented fields due to their operating time limitations and strong dependency on the traditional descriptors extracted from the image.

One of the most promising perspective in multimedia data search, storing and interpretation is to represent images as segmentation results and define metrics for their comparison such as Minkowski-type metric (including Euclidean and Manhattan distances), Mahalanobis metric, EMD, histogram metric, metric for probability density functions, sets of entropy metrics, pseudo metrics for semantic image classification (Rubner et al., 2000; Cheng et al., 2005; Wang et al., 2005). Yet, because of their limitations these metrics cannot give the desirable results, so a new metric was introduced and extended for considering the embedded partitions and it was effectively used for the content image retrieval (Kinoshenko et al., 2007). Due to the nested structure it becomes possible to perform the search with different level of refinement or roughening.

Using the region based image retrieval methods allows to make a step towards overcoming the semantic gap between law-level image description and high level conception. But from the other hand it definitely leads to increase of the computation complexity of image processing and distance calculation operations methods. Thus when creating an efficient indexing structure for image database one should first consider the methods providing minimal number of matching operations.

Many of multidimensional indexing methods used in the field of text retrieval were modified and improved in order to index high-dimensional image content descriptors. Among them X-trees, VA-file and I-Distance approaches are the most promising (Bohm et al., 2001). However, in case of comparing images as nested partitions there is no features to

describe complex objects and only information about distances between them is available, and so-called 'distance-based' indexing methods come to the aid (Chavez et al., 2001; Hjaltason, Samet, 2003). In this work existing 'distance-based' indexing methods are analyzed and improved and their possible application for the region-based image retrieval is considered.

Often on the pre-processing analysis stage we have only information of the mutual distance between the database objects. In this case such indexing methods as X-trees, VA-file, i-Distance, cannot be used as there is not enough information about objects coordinates. One of the possible solutions of this problem is to create an indexing structure based on the triangular inequality axiom. This principle lays in the base of metrical indexing methods. Here a distance matrix is formed and analysed for some selection of special objects subset (these objects are called pivots). Forming different data structures allows to eliminate from the consideration whole families of images situated far from the query at the search stage.

Clustering methods are often used for the image database preliminary processing. Thus we consider a possible hierarchical mechanism of the query processing: one can seek suitable clusters in nested partitions with an arbitrary indexing scheme. This way the amount of matches can be greatly reduced, but traditional clustering methods do not guarantee the optimal result.

Thus in order to optimize this Content-based image retrieval (CBIR) scheme it is necessary to minimize a total number of matches at the retrieval stage. We propose a new hierarchical clustering method which allows to construct images partitions into disjoint subsets so that firstly one can seek suitable class, then the most similar to the query subclass is chosen and so on. The exhaustive search is fulfilled only on the lowest level of hierarchy.

In this chapter we shall consider theoretical premises and methods of database images metrical indexing and find the tools providing guaranteed number of matching operations between a query and database objects.

2. Theoretical background for the CBIR distance matrix based indexing

Let $\mathrm{X} = \{x_1, x_2, ..., x_n\}$ be a set characterising images which constitute a database. Each element of this set can be:

- image itself $\mathrm{B}(z), z \in \mathrm{D} \subset \mathbb{R}^2$, where D is a sensor's field of view, $\mathrm{B}(z)$ is a brightness distribution function;
- feature vector $p \in \mathbb{R}^k$ $(\mathbb{Z}^k)$;
- some combination of the image processing results and features, for example segmentation results, contour preparations or regions shape features.

Then we consider $\mathrm{X} \subseteq \mathrm{U}$, where U is some universum which corresponds to an object-oriented field and provides introduction of the distance functional (metric in particular). Under such X we shall understand a database.

The task is to search the best suiting element (or elements) $x_i \in \mathrm{X}$, under given query $y \in \mathrm{U}$, which is represented (or can be brought to) by one of the listed above types. When we say "best suiting" we mean the minimal distance $\rho(y, x), y \in \mathrm{U}$.

It should be reminded that a non-negative function $\rho(y, x) \in \mathbb{R}^+$ is a metric on set U if

$\forall x, y, z \in \mathrm{U}$ next axioms are fulfilled:

a) $\rho(x, y) = 0 \Leftrightarrow x = y$ – reflexivity;

b) $\rho(x, x) = 0$ – symmetry;

c) $\rho(x, z) \le \rho(x, y) + \rho(y, z)$ – triangular inequality.

Talking about variety of image representations we need to emphasize that if there is a set of metrics $\rho_j(x, y)$, then their non-negative linear combination $\forall \alpha_j \ge 0 \ \sum_j \alpha_j \rho_j(x, y)$ is a metric. Moreover if some nonnegative function $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that $f(0) = 0$ and $f(z)$ is convex $\forall z \ge 0$, then $f(\rho(x, y))$ is a metric.

Using a metric for similarity criteria firstly provides adequacy of the search result to the query (if the choice of metric corresponding to the domain was correct), and secondly takes into account the triangular inequality, what creates premises for elimination from the consideration whole image sets without calculating distances to them. We shall note that search with limited matches number can be performed in two ways: using preliminary clustering in image or feature spaces, or considering methods which analyse values of pre-calculated distance matrix of all image collection elements. From the other hand all search algorithms can be divided into 3 groups:

- search of k most similar images ordered according to the similarity extent,
- search of the images which differ from the query on not more than given threshold δ,
- combination of these approaches.

Definition 1. (δ)- search result for the query $y \in \mathrm{U}$ is any element (all elements) $x_i \in \mathrm{X}$ if $\rho(y, x_i) \le \delta$ for given $\delta \ge 0$, which is called a search radius.

It is obvious that choice of δ threshold is a non trivial task. Moreover, choice of rational δ value essentially depends on configuration (mutual location in regard to the chosen metric) of the database objects. Still often the choice of this value is defined by the application, i.e. by the required similarity of the images. Note that under $\delta = 0$ we get a special case of searching the duplicates in the database.

Definition 2. Result of (k)- search for the query $y \in \mathrm{U}$ are elements of the set $\mathrm{X}^k = \{x_{i_1}, x_{i_2}, ..., x_{i_k}\} \subseteq \mathrm{X}$, for which

$$\forall x_{ij} \in \mathrm{X}^k, \forall x \in \mathrm{X} \setminus \mathrm{X}^k, \forall y \in \mathrm{U} \ \ \rho(y, x_{ij}) \le \rho(y, x), \ \ \rho(y, x_{ij}) \le \rho(y, x_{ij+1}), \ j = \overline{1, k-1}.$$

There is an important special case of $y \in \mathrm{X}, k = 1$. The exact match of the query should be found among the database elements, in other words the query should be identified and its corresponding image characteristics should be extracted. It is like identifying a person according to his finger prints.

Definition 3. The result of (δ, k)- search of query $y \in \mathrm{U}$ are elements of the set $\mathrm{X}^m = \left\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\right\} \subseteq \mathrm{X}$, $m \le n$, for which

$$\forall x_{i_j} \in \mathrm{X}^m,\ \forall x \in \mathrm{X} \setminus \mathrm{X}^m,\ \forall y \in \mathrm{U}\ \ \rho(y, x_{i_j}) \le \delta,\ \delta \ge 0\,,$$

$$\rho(y, x_{i_j}) \le \rho(y, x),\ \rho(y, x_{i_j}) \le \rho(y, x_{i_{j+1}}),\ j = \overline{1, m-1}\,.$$

The search is considered as successful one if there are elements satisfying definitions 1, 2 and 3. In other case a feedback is required to refine the query object, search parameters (for example radius δ), which is closely connected to the problem of representing images as features description and matching. It should be emphasised that formally (k)- search is always successful: decision as for query refining should be made on base of obtained distances analysis and the application requirements.

Calculation of N $\rho(y, x_i)$ values can be rather expensive, especially in the image space. Let us analyse the ways to reduce the operations number. For that a symmetric pair-distances matrix of database elements can be pre-calculated

$$d(\mathrm{X}) = \begin{pmatrix} 0 & \rho(x_1, x_2) & \rho(x_1, x_3) & \dots & \dots & \rho(x_1, x_n) \\ & 0 & \rho(x_2, x_3) & \dots & \dots & \rho(x_2, x_n) \\ & & 0 & \dots & \dots & \dots \\ & & & \dots & \dots & \dots \\ & & & & 0 & \rho(x_{n-1}, x_n) \\ & & & & & 0 \end{pmatrix}. \qquad (1)$$

Let $y \in \mathrm{U}$ be a query image. We shall fix some image $x^* \in \mathrm{X}$, called as a pivot object (point), and consider a triangular inequality picking out one more image $x_i \in \mathrm{X},\ i \in \{1, 2, ..., n\}$ (the distance $\rho(x^*, x_i)$ is known)

$$\rho(y, x_i) \le \rho(y, x^*) + \rho(x^*, x_i)\,, \qquad (2)$$

$$\rho(x_i, x^*) \le \rho(y, x^*) + \rho(y, x_i)\,, \qquad (3)$$

$$\rho(y, x^*) \le \rho(y, x_i) + \rho(x_i, x^*)\,. \qquad (4)$$

From equations (1) – (3) it follows that when knowing two distances, namely: $\rho(y, x^*)$ and $\rho(x^*, x_i)$, it is not hard to obtain upper and lower distance bounds

$$\left|\rho(x_i, x^*) - \rho(y, x^*)\right| \le \rho(y, x_i) \le \rho(y, x^*) + \rho(x_i, x^*)\,. \qquad (5)$$

Thus the implication takes place

$$\forall y \in \mathrm{U},\ \forall x_i, x^* \in \mathrm{X}\ :\ \rho(x_i, x^*) \ge 2\rho(y, x^*) \Rightarrow \rho(y, x_i) \ge \rho(y, x^*)\,. \qquad (6)$$

Let us consider the case when exact value of distance $\rho(x^*, x_i)$ is unknown but can be evaluated

$$\varepsilon_{min} \le \rho(x^*, x_i) \le \varepsilon_{max}\,. \qquad (7)$$

Then for x_i the following evaluation of the upper and lower bound of distance to y is obtained:

$$max\{\rho(y, x^*) - \varepsilon_{max}, \varepsilon_{min} - \rho(y, x^*), 0\} \le \rho(y, x_i) \le \rho(y, x^*) + \varepsilon_{max}. \quad (8)$$

Indeed, according to the triangular inequality and (7) we get

$$\rho(y, x^*) \le \rho(y, x_i) + \rho(x_i, x^*) \le \rho(y, x_i) + \varepsilon_{max},$$

then

$$\rho(y, x^*) - \varepsilon_{max} \le \rho(y, x_i). \quad (9)$$

From the other hand for objects x_i and x^* it is true that

$$\varepsilon_{min} \le \rho(x_i, x^*) \le \rho(y, x^*) + \rho(y, x_i),$$

from where

$$\varepsilon_{min} - \rho(y, x^*) \le \rho(y, x_i). \quad (10)$$

Expressions (9) and (10) are the lower bounds of $\rho(x, x_i)$, and to narrow the equation condition we chose the maximal value. Also both values can be negative simultaneously, what is shown in (8). Finally, evaluation of upper bound $\rho(y, x^*)$ directly follows from the triangular inequality (2) and condition $\rho(x^*, x_i) \le \varepsilon_{max}$.

It is easy to show that if the exact value $\rho(y, x^*)$ is unknown and within the limits

$$\sigma_{min} \le \rho(y, x^*) \le \sigma_{max}, \quad (11)$$

then the next evaluation of the lower and upper distance bound takes place:

$$max\{\sigma_{min} - \varepsilon_{max}, \varepsilon_{min} - \sigma_{max}, 0\} \le \rho(y, x_i) \le \sigma_{max} + \varepsilon_{max}. \quad (12)$$

From (8) and inequality $\sigma_{min} - \varepsilon_{max} \le \rho(y, x^*) - \varepsilon_{max}$, $\varepsilon_{min} - \sigma_{max} \le \varepsilon_{min} - \rho(y, x^*)$ the left part of (12) is obvious. The left part follows from $\rho(y, x^*) + \varepsilon_{max} \le \sigma_{max} + \varepsilon_{max}$ and right part of inequality (8).

Till now for evaluating $\rho(y, x_i)$ the distance from the target object x_i to pivot point x^* and distance from x^* to the search object y were considered. It is obvious that introducing to the analysis additional pivot point x_j^* in some cases will allow to narrow the ranking interval $\rho(x_i, y)$.

Let the object x_i be situated «closer» to the pivot point x_1^*, than to x_2^*, i.e.

$$\rho(x_i, x_1^*) \le \rho(x_i, x_2^*). \quad (13)$$

Then the equation takes place

$$max\left\{\frac{\rho(y,x_1^*)-\rho(y,x_2^*)}{2},0\right\}\le\rho(y,x_i)\,. \tag{14}$$

Indeed according to the triangular inequality axiom we have

$$\rho(y,x_1^*)\le\rho(y,x_i)+\rho(x_i,x_1^*)\,,$$

then

$$\rho(y,x_1^*)-\rho(y,x_i)\le\rho(x_i,x_1^*)\,. \tag{15}$$

Further the following takes place

$$\rho(x_i,x_2^*)\le\rho(y,x_2^*)+\rho(y,x_i)\,. \tag{16}$$

Using condition (13), from equations (15) and (16) we get

$$\rho(y,x_1^*)-\rho(y,x_i)\le\rho(y,x_2^*)+\rho(y,x_i)$$

what under possible negativity of expression $\frac{\rho(y,x_1^*)-\rho(y,x_2^*)}{2}$ gives evaluation (14).

3. Metrical search models

We shall consider some approaches to (δ)- search, which create the premises for creating an efficient indexing system to reduce a number of distance calculations on the search stage.
We shall agree that distance matrix $d(\mathrm{X})$ is calculated in the result of preliminary processing. The most simple but at the same time often most practically effective (in terms of matching operations number) indexing method is based on the full distance matrix calculation. Let us briefly describe this method. Suppose on iteration i there is a set of $\mathrm{X}^{(i-1)}\subseteq\mathrm{X}$ objects for which decision is not made if they are inside the search radius or not. It should be pointed out that $\mathrm{X}_0=\mathrm{X}$. Then some element $x^{(i-1)}\in\mathrm{X}^{(i-1)}$ is being chosen randomly or according to some criteria, $\rho(y,x^{(i-1)})$ is calculated and the next set is produced

$$\mathrm{X}^{(i)}=\{x_j^{(i-1)}\in\mathrm{X}^{(i-1)}\bigcap\{x^{(i-1)}\}:|\rho(x^{(i-1)},x_j^{(i-1)})-\rho(y,x^{(i-1)})|\le\delta\}\,. \tag{17}$$

It is obvious that $\forall x'^{(i-1)}_j\in X^{(i-1)}\bigcap X^{(i)}\Rightarrow|\rho(x^{(i-1)},x_j^{(i-1)})-\rho(y,x^{(i-1)})|>\delta$ and all such elements can be eliminated from the consideration. Figure 1 illustrates forming of set $\mathrm{X}^{(1)}$ on first iteration: all elements $x_i\in\mathrm{X}^{(0)}$ situated in the crosshatched region are eliminated from consideration and the rest of elements will form set $\mathrm{X}^{(1)}$.

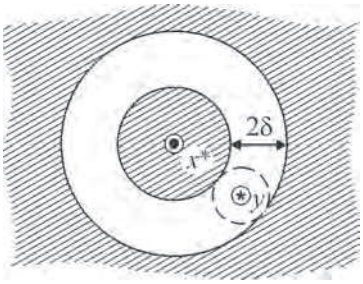


Fig. 1. Geometrical interpretation of eliminating the elements based on the lower distance evaluation

Let Y be a goal set of elements X, situated inside of the search radius, which is empty on the initial stage. Then on i iteration we get $\mathrm{Y} = \mathrm{Y} \bigcup x^{(i-1)}$, if $\rho(y, x^{(i-1)}) \le \delta$. The procedure is carried out recursively till step l, when

$$\forall k = \{l-m, l-m+1, \ldots, l\} \;\; card(\mathrm{X}^{(k-1)}) - 1 = card(\mathrm{X}^{(k)}),\; \mathrm{X}^{(k)} \subseteq \mathrm{X}^{(k-1)},$$

i.e. exactly one element is eliminated from consideration during m iterations. In this case $\rho(x_j^{(l)}, y)$, $x_j^{(l)} \in \mathrm{X}^{(l)}$, $j = \overline{1, card(X^{(l)})}$ are calculated and evaluated directly completing the Y forming. Total number of matching operations will be equal to

$$\mathrm{N}(\delta) = l + card(\mathrm{X}^{(l)}). \tag{18}$$

In practice storing full distance matrix $d(\mathrm{X})$ is expensive due to the quadratic dependency of the required memory on database dimensionality. To solve this problem which is extremely acute on big data sets, a 'sparse' view $d(\mathrm{X})$ is used or special data structures are formed which in some way approximate the distance matrix considering correlation of its elements. Obviously a natural requirement for such methods would be a compromise between cost of resources storage and distance matching operations number at the search stage which should tend to $\mathrm{N}(\delta)$ in (18). At the same time it should be noted that value $\mathrm{N}(\delta)$ is random in sense that it depends on the objects space configuration, mutual location and order of choosing objects $x^{(1)}, x^{(2)}, \ldots, x^{(l)}$, as well as location of query object y. For example, $\mathrm{N}(\delta)$ value can be decreased if object $x^{(2)}$ is chosen at first iteration, and $x^{(1)}$ is chosen at the second one. Thus indexing methods which operate on not complete information comparing with method on complete distance matrix theoretically can perform less matching operation then the last one.

Let us introduce a set of fixed image database elements $\mathrm{X}^* = \{x_1^*, x_2^*, ..., x_k^*\}$. From (5) the lower distance bound follows

$$\rho(y, x_i) \geq \rho_{X^*}(y, x_i) \tag{19}$$

where $\rho_{X^*}(y, x_i) \triangleq \max_{x^* \in X^*} |\rho(y, x^*) - \rho(x^*, x_i)|$. It is the simplest indexing method based on a 'sparse' distance matrix where $d(X)$ after corresponding re-ordering of indexes takes the form

$$d(X)_k^* = \begin{pmatrix} 0 & 0 & 0 & \dots & \rho(x_1, x_{k+1}) & \dots & \rho(x_1, x_n) \\ & 0 & 0 & \dots & \rho(x_2, x_{k+1}) & \dots & \rho(x_2, x_n) \\ & & 0 & \dots & \dots & \dots & \dots \\ & & & 0 & \rho(x_k, x_{k+1}) & \dots & \rho(x_k, x_n) \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} \rho(x_1, x_{k+1}) & \dots & \rho(x_1, x_n) \\ \rho(x_2, x_{k+1}) & \dots & \rho(x_2, x_n) \\ \dots & \dots & \dots \\ \rho(x_k, x_{k+1}) & \dots & \rho(x_k, x_n) \end{pmatrix}. \tag{2.20}$$

It should be noted that this approach can be interpreted as mapping $(X, \rho) \to (\mathbb{R}^k, \rho_\infty)$ and fulfilling the search in k-arity space.

Lower distance bounds (8), (12), (14) can be obtained without information about exact $\rho(y, x^*)$ or $\rho(x^*, x_i)$ values. For example, m columns of some rows of the sparse distance matrix can be defined by interval (7). If $\varepsilon_{max} - \varepsilon_{min} \to 0$, lower bound (8) will tend to the lower bound (5). Thus, the greater is m value, the greater is the volume of released memory, which can be used for the additional pivot objects choice what should potentially increase indexing algorithms efficiency, for example will increase the $\rho_{X^*}(y, x_i)$ value for the sparse distance matrix. Further let for each non-pivot object x_i be defined the closest in terms of chosen metric pivot object x^*, what allows to form a membership matrix. Then on the search stage for obtaining the lower distance bound $\rho(y, x_i)$ one can use (14) and there is no necessity for storing exact values $\rho(x_i, x^*)$. It can be noted that produced membership matrix is few times more compact (depending on programming realization) as for the resources used for its storing comparing with the distance matrix. All that also allows to allocate greater number of pivot objects.

The described approaches allow to construct different variants of data structures, including hierarchical ones for the indexing organization. We will consider several of them.

We shall chose a random point x^* and calculate the distance $\rho(x^*, x_i), x_i \in X$. We shall introduce the equivalence relation φ_ρ such that $x_i \underset{\varphi_\rho}{\sim} x_j$, if $\rho(x^*, x_i) = \rho(x^*, x_j)$. Given relation allows to partite X into equivalence classes $\{X_d\}$, where $X_d = \{x_i \in X : \rho(x^*, x_i) = d\}$ for $d > 0$, $X_0 = \{x^*\}$. The partition process can be continued recursively inside of each class. On the search stage all X_d, for which $|\rho(y, x^*) - d| > \delta$ is fulfilled are eliminated from consideration.

Let us consider another method of X partition. We shall call it a binary tree method. On the starting stage a pivot object x_j^*, which has index j in the matrix $d(\mathrm{X})$ is singled out, and row j of the given matrix is calculated. This way we shall define distances to all other objects $d(\mathrm{X})_{j,1}, d(\mathrm{X})_{j,2}, \ldots, d(\mathrm{X})_{j,n}$. We shall sort values of row j ascending, having re-labelled indices $d^*(\mathrm{X})_{j,1}, d^*(\mathrm{X})_{j,2}, \ldots, d^*(\mathrm{X})_{j,n}$, and define $\mathrm{M} = \rho(x_j^*, d^*(\mathrm{X})_{j,\left\lceil \frac{n}{2} \right\rceil})$ – the distance to the median object. We produce partition of X into two classes $\mathrm{X}_{\leq} = \{ x_i \in \mathrm{X} : \rho(x_j^*, x_i) \leq \mathrm{M} \}$ and $\mathrm{X}_{>} = \{ x_i \in \mathrm{X} : \rho(x_j^*, x_i) > \mathrm{M} \}$ (Figure 2).

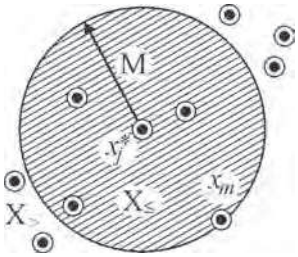
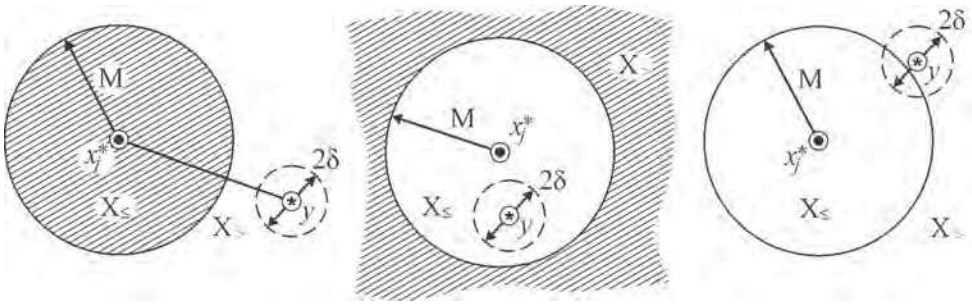


Fig. 2. Geometrical interpretation of the indexation method structure forming by binary tree

On the search stage it is necessary to search only class $\mathrm{X}_{\leq}$ under $\rho(x_j^*, y) \leq \mathrm{M} - \delta$, and class $\mathrm{X}_{>}$ under $\rho(x_j^*, y) > \mathrm{M} + \delta$. On figure 3-a is shown that when the first equation is fulfilled $\mathrm{X}_{>}$ contains the hypersphere formed around the query object. Figure 3-b illustrates fulfilment of the second equation.

Thus when one of two mentioned equations is fulfilled a half of the elements in set X can be eliminated from the consideration. If not, i.e. when $\mathrm{M} + \delta \leq \rho(x_j^*, y) \leq \mathrm{M} - \delta$, hypersphere with centre in x_j^* and radius M intersects hypersphere of the query object, analysis of each of the $\mathrm{X}_{\leq}$ and $\mathrm{X}_{>}$ sets is required (figure 3-c).

One possible way to solve the problem is to introduce into consideration an additional equivalence class $\mathrm{X}_{\mathrm{M},\lambda} = \{ x_i \in \mathrm{X} : \mathrm{M} - \lambda < \rho(x_j^*, x_i) \leq \mathrm{M} + \lambda \}$ and redefine existing classes $\mathrm{X}_{\leq} = \{ x_i \in \mathrm{X} : \rho(x_j^*, x_i) \leq \mathrm{M} - \lambda \}$ and $\mathrm{X}_{>} = \{ x_i \in \mathrm{X} : \rho(x_j^*, x_i) > \mathrm{M} + \lambda \}$, where $\lambda > \delta$.

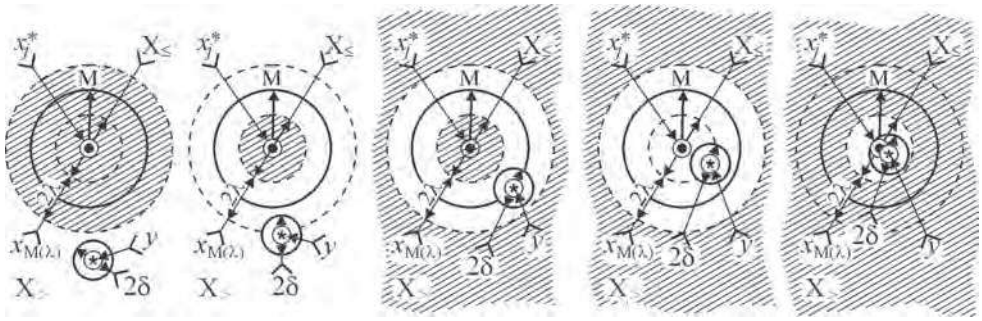


a) b) c)

Fig. 3. Three possible variants of considering sets $X_>$, $X_\leq$ under different location of the query object

In this case on the search stage (depending on the data configuration and query object location) consideration of the following classes will be required (see figure 4 where all elements situated in the crosshatched region are eliminated from the consideration):

- $X_>$ under $\rho(y,x_j^*) > M+\lambda+\delta$ (figure 4-a);
- $X_>$ and $X_{M,\lambda}$ under $M+\lambda-\delta < \rho(y,x_j^*) \leq M+\lambda+\delta$ (figure 4-b);
- $X_{M,\lambda}$ under $M-\lambda+\delta < \rho(y,x_j^*) \leq M+\lambda-\delta$ (figure 4-c);
- $X_{M,\lambda}$ and $X_\leq$ under $M-\lambda-\delta < \rho(y,x_j^*) \leq M-\lambda+\delta$ (figure 4-d);
- $X_\leq$ under $\rho(y,x_j^*) \leq M-\lambda-\delta$ (figure 4-e).



a) b) c) d) e)

Fig. 4. Possible variants of considering sets $X_>$, $X_{M,\lambda}$, $X_\leq$ under different location of the query object

Obtained partition has both advantages and disadvantages. Obviously regardless of the location of y one of sets $X_{\leq}$, $X_{>}$ will be eliminated from the consideration. From the other hand cardinalities of $X_{\leq}$ and $X_{>}$ can differ and thus the search tree will not be balanced.

Partition of set X should be «granulated» recursively under $card(X) > b$, where b is maximally allowed cardinality of «leaf» set, $b \geq 3$.

Let us consider method of indexing by partitions. We shall fix a set of pivot objects $X^* = \{ x_1^*, x_2^*, \ldots, x_k^* \}$. If elements x_s^*, $s = \overline{1,k}$ form regions X_s^* based on the following

$$X_s^* = \{ x_i \in X : \forall t = \overline{1,k}, t \neq s \ \rho(x_i, x_s^*) < \rho(x_i, x_t^*) \}, \tag{21}$$

then $\{X_s^*\}$ is a partition of X. Notice that in case when metrical space is a $\mathbb{R}^2$, partition elements are cells of the Voronoi diagram. Voronoi diagram for the finite points set S on plane is a partition of this plane in such way that each region of this partition forms a set of points which are more close to one of the elements of set S, than to any other element of the set.

Introduced in this way partition allows to obtain several lower bounds of distance $\rho(y, x_i)$. We shall denote them as $\rho_{min}^{(p)}$, where p is a number of evaluation.

Condition (13) is fulfilled in consequence of (21), as $\rho(x_i, x_s^*) \leq \rho(x_i, x_t^*) \ \ \forall x_i \in X_s^*$, $t \neq s$. Then from (14) for k pivot points we get evaluation of lower bound of distance $\rho(y, x_i)$ $\forall x_i \in X_s^*$:

$$\rho_{min}^{(1)}(s) = max\left\{ \max_{t=\overline{1,k}, t \neq s} \left\{ \frac{\rho(y, x_s^*) - \rho(y, x_t^*)}{2} \right\}, 0 \right\}. \tag{22}$$

Let $\varepsilon_{max}(s) = \max\limits_{x_i \in X_s^*} \rho(x_s^*, x_i)$ be a radius of partition X_s^* cover. Then according to (8) we get second lower distance evaluation:

$$\rho_{min}^{(2)}(s) = \rho(y, x_s^*) - \varepsilon_{max}(s). \tag{23}$$

On the preliminary processing stage minimal and maximal distances between partitions should be calculated:

$$\varepsilon_{min}(s,t) = \min_{x_i \in X_t^*} \rho(x_s^*, x_i), \tag{24}$$

$$\varepsilon_{max}(s,t) = \max_{x_i \in X_t^*} \rho(x_s^*, x_i), \tag{25}$$

where $s, t = \overline{1,k}$.

Then $\varepsilon_{min}(s,t)$ and $\varepsilon_{max}(s,t)$ under $s \neq t$ are corresponding evaluations of ε_{min} and ε_{max} distances $\rho(x_s^* x_i)$ $\forall x_i \in X_t^*$ in (8). It should be noted that under $s = t$ $\varepsilon_{min}(s,s) = 0$

and $\varepsilon_{max}(s,s) = \max\limits_{x_i \in X_s^*} \rho(x_s^*, x_i) = \varepsilon_{max}(s)$. From here it can be stated that evaluation $\rho_{min}^{(2)}(s)$ in (23) is a special case of evaluation

$$\rho_{min}^{(3)}(s) = max\{\max\limits_{t=1,k}\{\rho_{min}^{(3)}(s,t)\}, 0\}, \tag{26}$$

$$\rho_{min}^{(3)}(s,t) = max\{\rho(y,x_s^*) - \varepsilon_{max}(s,t), \varepsilon_{min}(s,t) - \rho(y,\rho(y,x_s^*))\}. \tag{27}$$

Evaluation $\rho_{min}^{(1)}(s)$ allows to eliminate from the consideration all elements of class X_s^*, while evaluation $\rho_{min}^{(3)}(s,t)$ allows to obtain lower distance bound for class (and all its objects) X_t^* with no necessity for calculating the distance $\rho(y,x_t^*)$.
Finally, the maximal lower bound of distance $\rho(y,x_i)$, $x_i \in X_s^*$ is defined as

$$\rho_{min}(s) = max\{\rho_{min}^{(1)}(s), \rho_{min}^{(3)}(s)\}. \tag{28}$$

One way to use the obtained distance lower bounds on the search stage is to calculate k distances to pivot objects and eliminate those classes X_s^*, for which $\rho_{min}(s) > \delta$ is fulfilled. This task can be solved more optimally by defining the distance from y to not all of elements x_s^*. Let E contain a set of pivot objects producing the partitions which cannot be eliminated. On the initial stage $E = \{x_s^*\}_{s=1}^k$. Then we chose random $x_s^* \in E$, for which $\rho(y,x_s^*)$ is not calculated, compute this distance and fix lower bound $\rho_{min}^{(3)}(s,t)$ on (2.27) for all $x_t^* \in E$, performing $E = E \bigcap \{x_t^*\}$ in case of $\rho_{min}^{(3)}(s,t) > \delta$. The procedure is repeated till E becomes an empty set or till distances to all $x_s^* \in E$ are calculated. As interval $(\varepsilon_{min}(s,t), \varepsilon_{max}(s,t))$ can be large, the refinement of the lower bound of the regions still contained in E can be done via (22).
The algorithms described above do not exhaust all the potentialities of the distance matrix analysis. The matrix factorization provides a certain perspective for pivot points choice and the search itself. This factorization in fact correspons to object clustering in the given metrical space. Let us consider such a method.

4. Block-diagonal forms of distance matrix

We shall denote $\rho_{i,j} = \rho(x_i, x_j)$ as elements of distance matrix $d(X)$ and consider a random subset of the database objects for which $\rho_{i,j} \le \delta$. These objects can be used as the result of (δ)- search, although for that the formalization of search of all such groupings under given criteria is needed. Here a number of matching operations between query and database objects will be taken as a main criteria. In other words the task lies in preliminary

clustering of the database with the search strategy of finding the closest cluster and if necessary continue the search inside of the chosen cluster. The goal of the clustering here is minimization of the matching operations number (under given precision of search).

Definition 4. We shall name a quadratic symmetric matrix of l-th order as Δ_l^k-block of a distance matrix $d(\mathrm{X})$

$$\Delta_l^k[d(\mathrm{X})] = \begin{pmatrix} 0 & \rho_{k,\,k+1} & \rho_{k,\,k+2} & \cdots & \rho_{k,\,k+l-1} \\ \rho_{k+1,\,k} & 0 & \rho_{k+1,\,k+2} & & \rho_{k+1,\,k+l-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{k+l-1,\,k} & \rho_{k+l-1,\,k+1} & \cdots & \rho_{k+l-1,\,k+l-2} & 0 \end{pmatrix}$$

which is the result of rearranging of rows and columns with numbers $\{i_1, i_2, ..., i_l\}$ such that $\forall i',\ i'' \in \{i_1, i_2, ..., i_l\} \Rightarrow \rho_{i'i''} \le \delta$.

We shall call a Δ_l^k-block of matrix $d(\mathrm{X})$ maximal if

$$\nexists r \in \{1, 2, ..., n\} \setminus \{i_1, i_2, ..., i_l\} : \rho_{r,\,i'} \le \delta \quad \forall i' \in \{i_1, i_2, ..., i_l\}.$$

It should be emphasised that some elements can belong to two or more Δ_l^k-blocks of matrix $d(\mathrm{X})$ simultaneously. Obviously there are two possible variants in this case: they are included into all possible blocks, or they are included into blocks where sums of elements are minimal, what meets the maximal clusters compactness criterion. Generally in the first case we deal with a cover (weak clustering) of database elements set, in the second case – with partition. In result the means of Δ_l^k-blocks forming influences the search algorithms implementation, especially if more detailed result is required, for instance for (k)- or (δ, k)- search.

We shall call a Δ-representation a block-diagonal shape of matrix $d(\mathrm{X})$

$$\Delta[d(\mathrm{X})] = \begin{pmatrix} \Delta_{l_1}^{k_1} & & & 0 \\ & \Delta_{l_2}^{k_2} & & 0 \\ 0 & & \cdots & 0 \\ & 0 & 0 & \Delta_{l_m}^{k_m} \end{pmatrix},$$

where $k_1 = 1,\ k_i = \sum_{j=1}^{i-1} l_j + 1,\ \sum_{j=1}^{m} l_j \ge n$.

It is clear that the best Δ-representation (in sense of matching operations number) at (δ)- search, is the one with minimal number of blocks. In other words forming of matrix Δ-representation under given δ should provide

$$\min_{\Delta_l^k \in d(\mathrm{X})} m. \tag{29}$$

In the situation mentioned above the criterion can be considered

$$\min_{\Delta_l^k \in d(\mathrm{X})} \sum_{i,j \in \{i_1, i_2, \ldots, i_l\}} \rho_{i,j} \tag{30}$$

which does not change the value of goal function (29), but allows to obtain more adequate clustering of the database.

We shall consider the procedure of forming the maximal Δ_j^i-block of distance matrix $d(\mathrm{X})$ on set $\{p_1, p_2, \ldots, p_r\} \subseteq \{1, 2, \ldots, n\}$. We shall find a new row α^* such that

$$\alpha^* = arg \max_{\alpha \in \{p_1, p_2, \ldots, p_r\}} \{card\{\rho_{\alpha,q}\} : \rho_{\alpha,q} \le \delta,\ q \in \{p_1, p_2, \ldots, p_r\}\}. \tag{31}$$

Let us discuss the choice of one of them (under equality α^* for some rows). We denote found in (31) indices $q \in \{p_1, p_2, \ldots, p_r\}$ as $\{\alpha_1, \ldots, \alpha^*, \ldots, \alpha_\beta\}$. There also can be few of such sets: let us chose any of them. Two cases are possible:

$$\forall \alpha', \alpha'' \in \{\alpha_1, \ldots, \alpha^*, \ldots, \alpha_\beta\} \Rightarrow \rho_{\alpha', \alpha''} \le \delta, \tag{32}$$

$$\exists \alpha', \alpha'' \in \{\alpha_1, \ldots, \alpha^*, \ldots, \alpha_\beta\} \text{ such that } \rho_{\alpha', \alpha''} > \delta. \tag{33}$$

Implication (32) means that a choice of α^* guarantees forming of maximal $\Delta_{\beta+1}^i$-block of distance matrix $d(\mathrm{X})$ on set $\{p_1, p_2, \ldots, p_r\}$. So, having redefined the search field

$$\{p_1, p_2, \ldots, p_r\} \leftarrow \{p_1, p_2, \ldots, p_r\} \setminus \{\alpha_1, \alpha_2, \ldots, \alpha_\beta\}, \tag{34}$$

if $\{p_1, p_2, \ldots, p_r\} \neq \varnothing$, we can move to the construction of maximal Δ_j^i-block, starting with (31). This case is illustrated in figure 5.

Situation described in (33) is more complicated, but it can be brought to (32) by sequential elimination of far situated elements. Here 2 situations are possible as well: equality of the eliminated and remained elements and their discrepancy.

In case of the elements count discrepancy, till fulfilment of (32) every element α'_γ, ($\gamma = 1, 2, \ldots, \Gamma$, $\{\alpha'_0\} = \varnothing$, Γ: $\exists \alpha', \alpha'' \in \{\alpha_1, \ldots, \alpha_\beta\} \Rightarrow \rho_{\alpha', \alpha''} > \delta$, are sequentially eliminated $\{\alpha_1, \ldots, \alpha_\beta\} \leftarrow \{\alpha_1, \ldots, \alpha_\beta\} \setminus \{\alpha'_{\gamma-1}\}$) such that

$$\alpha'_\gamma = arg \max_{s \in \{\alpha_1, \ldots, \alpha_\beta\}} \{card\{\rho_{q,s}\} : \rho_{q,s} > \delta,\ q \in \{\alpha_1, \ldots, \alpha_\beta\}\}. \tag{35}$$

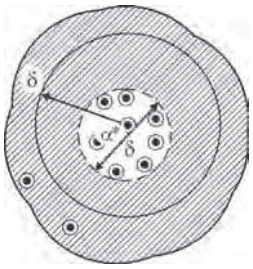


Fig. 5. Geometrical interpretation of Δ_j^i -block under (32)

If cardinality of reduced indexing set exceeds the number of close elements of the next biggest on (31) value distance matrix row it means that the next block is obtained. Otherwise, having temporarily eliminated α'_γ from $\{p_1, p_2, ..., p_r\}$, we repeat the considered steps till the next Δ_j^i -block is obtained, after that all eliminated rows are brought back for further analysis. Figure 6-a illustrates geometrical interpretation of this case, figure 6-b illustrates the result of Δ_j^i -blocks forming.

If at any step of the procedure there is an equal number of the eliminated and remained elements (figure 7) element α^*, as was mentioned above, can be included into all blocks (on the given step we get a few maximal blocks and reduction (34) is fulfilled multiple times) or using (30), we chose a maximal compact block, and after that start the formation of the next block.

Now we can discuss the choice of α^* when having multiple rows in (31). First of all we shall emphasise that for (32) we have exactly β rows and choice of α^* does is of no importance, as all elements will be simultaneously put into one Δ_j^i -block (see figure 5), or there are few blocks of the same cardinality, and all of them are obtained sequentially. If the (33) takes place, then

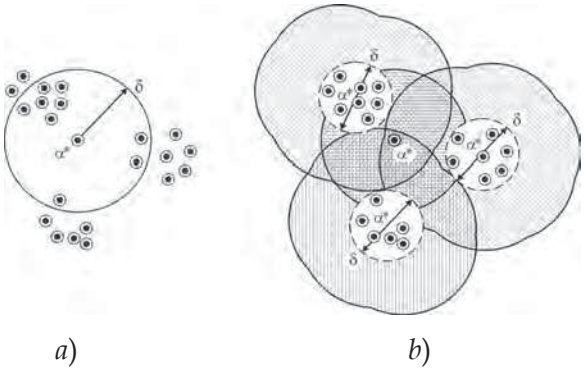


Fig. 6. Geometrical interpretation of -blocks forming for (33)

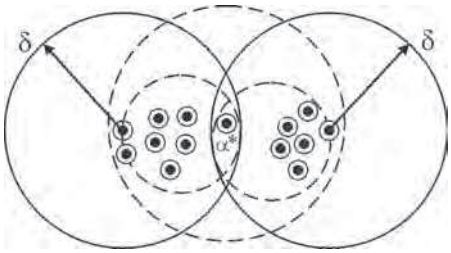


Fig. 7. Forming of Δ_j^i-blocks when the number of eliminated and remained elements are equal

the choice of α^* is also random, as accurate within the numeration all maximal blocks will be sequentially reduction formed till cases (32).

Thus, we have considered forming of some Δ_j^i-block on index set $\{p_1, p_2,..., p_r\} \subseteq \{1, 2,..., n\}$. After search and renumbering on set $\{1, 2,..., n\}$ of first maximal block in Δ-representation of distance matrix we get

$$\begin{pmatrix} 0 & \rho_{1,2} & \rho_{1,3} & \cdots & \rho_{1,k_1} & & & & & & \\ \rho_{2,1} & 0 & \rho_{2,3} & \cdots & \rho_{2,k_1} & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & & & & & & \\ \rho_{k_1,1} & \rho_{k_1,1} & \cdots & \rho_{k_1,k_1-1} & 0 & & & & & & \\ & & & & & & & 0 & & & \\ & & & & & 0 & \rho_{k_1+1,k_1+2} & \rho_{k_1+1,k_1+3} & \cdots & \cdots & \rho_{k_1+1,n} \\ & & & & & \rho_{k_1+2,k_1+1} & 0 & \rho_{k_1+2,k_1+1} & \cdots & \cdots & \rho_{k_1+2,n} \\ & & 0 & & & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & \rho_{k_1+n-1,k_1+1} & \rho_{k_1+n-1,k_1+2} & \cdots & \cdots & 0 & \rho_{k_1+n-1,n} \\ & & & & & \rho_{n,k_1+1} & \rho_{n,k_1+2} & \cdots & \cdots & \rho_{n,n-1} & 0 \end{pmatrix}.$$

Further elimination of indices set $\{1, 2,..., k_1\}$, we get distance matrix of dimension $(n-k_1)\times(n-k_1)$. Repeating the procedure for every new distance matrix we at first get the required representation, at second due to maximality of blocks on each steps fulfil (29).

Some notes should be made. First of all, it is not necessary to eliminate elements α'_γ, as the next element α^* can be temporary omitted. Maximal Δ_j^i-block will still be found, but possibly it will take more iterations to execute. It can be emphasised that efficiency of computational models of forming Δ-representations will improve a lot if rows of matrix $d(\mathrm{X})$ is ordered descending accordingly to (31) criterion. Firstly reduction (34) will be simplified due to sequential elimination of the distance matrix rows. Secondly, in many cases we will not need renumbering of Δ_j^i-block elements under including it into the Δ-representation.

5. Computational models optimization.

In practice depending on data configurations used in the metric space fulfilment of (29) may not provide sufficient search efficiency. It will happen if for example value δ was not chosen rationally, or dimensionalities of blocks in Δ-representations are still too large to be used as a result of the (δ)- search. Then, one way is to keep searching inside of the best block (best in terms of closeness to the query), another way is to expand (extend) the preliminary processing. Particularly within distance matrix preprocessing Δ-representations can be formed in each of the found Δ_j^i -blocks. Though the task becomes more complicated as it is not known which of the Δ-representations will be used on the second search stage. In other words the guaranteed search operations number is determined by number of Δ_j^i -blocks, obtained on the first stage, maximal number of nested $\Delta_{j'}^{i'}$ -blocks in each of Δ_j^i -blocks and maximal number of their elements. At that as Δ-representation with closeness parameter δ is already found, we can confine ourselves to splitting on nested Δ_j^i -blocks of possibly same dimension, maximising intra-cluster distances.

We shall order obtained Δ_j^i -blocks ascending their dimensions. Let us have s_0 elements not included into any of the blocks, s_1 blocks of dimension l_1, s_2 – of dimension $l_2,\ldots,s_t$ s_2 – of dimension l_t, i.e. $1=l_0 \le l_1 < l_2 < l_t \le n$. Then, assuming that we need to split blocks starting with dimension l_i, and denoting the maximal dimension of the resulting blocks as $\mathrm{M} \in [l_{i-1}, l_i] \cap \mathbb{N}$ (figure 8), we get two possible search realization strategies. The first one is searching the best block of nested Δ-representation, then the nested block of Δ-representation and exhaustive search in the closest Δ_j^i -blocks. The second one consists in choosing the block among union of all blocks of two-level Δ- representation of the distance matrix. The first strategy is a special case of the stratificated analysis and is out of our research field. Let us stop on the second one: maximal matching operation number is equal to the sum of blocks number and their maximal dimensionality, i.e.

$$f(\mathrm{M}) = \mathrm{M} + \sum_{j=1}^{i-1} s_j + \sum_{j=i}^{t} \left\lceil l_j / \mathrm{M} \right\rceil s_j,\ i = \overline{1,t} \tag{36}$$

where $\lceil \circ \rceil$ is minimal integer number exceeding the given one.

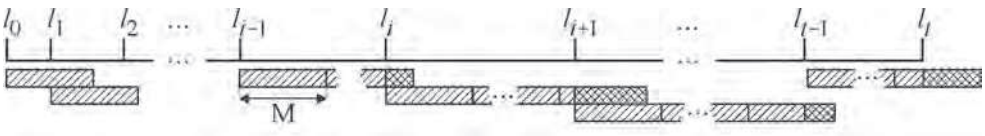


Fig. 8. Splitting of Δ_j^i -blocks