I T H E A

# International Journal

# INFORMATION THEORIES & APPLICATIONS

2009 Volume 16 Number 3

International Journal
INFORMATION THEORIES & APPLICATIONS
Volume 16 / 2009, Number 3



IJ ITA is official publisher of the scientific papers of the members of the ITHEA® International Scientific Society

IJ ITA welcomes scientific papers connected with any information theory or its application.

IJ ITA rules for preparing the manuscripts are compulsory.
The rules for the papers for IJ ITA as well as the subscription fees are given on *www.ithea.org* .
The camera-ready copy of the paper should be received by http://ij.ithea.org.
Responsibility for papers published in IJ ITA belongs to authors.

General Sponsor of IJ ITA is the Consortium FOI Bulgaria (www.foibg.com).

International Journal "INFORMATION THEORIES & APPLICATIONS" Vol.16, Number 3, 2009

Printed in Bulgaria

Edited by the Institute of Information Theories and Applications FOI ITHEA®, Bulgaria,
in collaboration with the V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
and the Institute of Mathematics and Informatics, BAS, Bulgaria.

Publisher: ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com



ISSN 1310-0513 (printed) ISSN 1313-0463 (online) ISSN 1313-0498 (CD/DVD)

# DISTANCE MATRIX APPROACH TO CONTENT IMAGE RETRIEVAL

Dmitry Kinoshenko, Vladimir Mashtalir, Elena Yegorova

*Abstract: As the volume of image data and the need of using it in various applications is growing significantly in the last days it brings a necessity of retrieval efficiency and effectiveness. 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. To meet higher requirements, a novel distance-based indexing method for region-based image retrieval has been proposed and investigated. The method creates premises for considering embedded partitions of images to carry out the search with different refinement or roughening level and so to seek the image meaningful content.*

*Keywords: content image retrieval, distance matrix, indexing.*

*ACM Classification Keywords: H.3.3 Information Search and Retrieval: Search process*

## Introduction

For the image retrieval from large scale database traditionally queries 'ad exemplum' are used. There are many approaches developed considering similarities of the query and images in database based on the distance between feature vectors which contain image content descriptors, such as color, texture, shape, etc [Greenspan et al., 2004; Yokoyama, Watanabe, 2007]. The most effort at present is put on the problem of putting the image retrieval on a higher semantic level to perform the search based on the image meaningful content, and not just on image own properties.

There are many metrics developed and widely used in image processing applications: 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, by virtue of their limitations these metrics cannot give the desirable results, so a new metric was introduced and extended for considering the embedded partitions and so it was effectively used for the content image retrieval [Kinoshenko et al., 2007]. Due to the embedded structure it will become possible to perform the search with different level of refinement or roughening.

As volume of multimedia databases grows exponentially a great need of means of fast search arises. 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 l-Distance approaches are the most promising [Bohm et al., 2001]. However, in case of comparing images as embedded partitions we do not have the 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.

## Theoretical background of distance matrix based content image retrieval

Let $X = \{x_1, x_2, ..., x_n\}$ be a set characterizing images in the database. Each element of this set can represent as an image $B(z), z \in D \subset R^2$ ( $D$ is a sensor field of view); as feature vector $p \in R^k$ ($Z^k$) ; as some combination of image processing results and features (e.g. segmentation, detected edges, shape features).

Further, keeping the generality of consideration, $X \subseteq U$ where $U$ is some universum ensuring introducing of similarity functional (specifically metric), we shall understand as an image database putting aside the indexing. The task consists in the search for correspondence of elements $x_i \in X$ in the best way to the query $y \in U$. Under 'the best way' we shall understand the minimum of distance $\rho(y, x), y \in U$, $x \in X$. Using metric as a similarity criterion provides adequacy of the search result to the query and the triangular inequality makes premises for excluding from the consideration whole sets of images without calculating distances to them.

We shall note, that there are 2 ways to perform the search with limited matches number: either by using preliminary clustering in image or feature spaces, or based on methods analyzing values of pre-calculated distance matrix for all images collection elements. Ex altera parte, all search algorithms can be classified as follows: search of $k$ most similar images ordered according to their similarity; search of the images which differ from the query on not more than given $\delta$ (range queries), and combination of these two approaches.

Definition 1. The result of $(\delta)$- search on query $y \in U$ is any element (all the elements) $x_i \in X$, if $\rho(y, x_i) \le \delta$ for given $\delta \ge 0$, called as the search radius.

It is clear that choice of range $\delta$ is a non-trivial task. Moreover, choice of rational value $\delta$ much depends on the database objects configuration (mutual location regarding the chosen metric). However often the choice of this value is dictated by the practical application, i.e. required extent of image similarity.

Definition 2. The result of $(k)$- search on query $y \in U$ are elements of set $X^k = \{x_{i_1}, x_{i_2}, ..., x_{i_k}\} \subseteq X$, for which $\forall x_{i_j} \in X^k, \forall x \in X \setminus X^k, \forall y \in U \ \ \rho(y, x_{i_j}) \le \rho(y, x), \ \rho(y, x_{i_j}) \le \rho(y, x_{i_{j+1}}), \ j = \overline{1, k-1}$.

It is necessary to indicate a rather important special case: $y \in X, k = 1$. It means that it is needed to find exact coincidence of query with a database element, i.e. practically identify query and detect image characteristics connected to it, e.g. to identify a person according to his fingerprints.

Definition 3. The result of $(\delta, k)$-search on query $y \in U$ are elements of set $X^m = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\} \subseteq X$, $m \le n$, for which $\forall x_{i_j} \in X^m, \forall x \in X \setminus X^m, \forall y \in U, \rho(y, x_{i_j}) \le \delta \ \Rightarrow \rho(y, x_{i_j}) \le \rho(y, x), \rho(y, x_{i_j}) \le \rho(y, x_{i_{j+1}}), j = \overline{1, m-1}$.

We shall call a search successful if there are elements satisfying definitions 1, 2 and 3. Otherwise, the feedback coupling is needed i.e. query object or search parameters (for instance radius $\delta$) refinement what is closely connected to the image presentation by feature descriptions and their matching. We shall emphasize that formally $(k)$- search is always successful as query refinement decision should always being made on the base of obtained distances analysis and solving task requirements. Notice that each successive search type is more

complicated to solve than the previous one. Thus procedures of handling the $(\delta)$- search are to be used as pre-processing during $(k)$- search, $(\delta, k)$- search should exploit $(\delta)$- search and $(k)$- search results.

We shall analyze the possibility of reducing the number $N$ of calculative values $\rho(y, x_i)$ which can be rather computationally expensive especially in the image space. With that purpose a symmetrical matrix of all pairwise distances of all database elements

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

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

$$\rho(y, x_i) \le \rho(y, x^*) + \rho(x^*, x_i) \tag{2}$$

$$\rho(x_i, x^*) \le \rho(y, x^*) + \rho(y, x_i) \tag{3}$$

$$\rho(y, x^*) \le \rho(y, x_i) + \rho(x_i, x^*) \tag{4}$$

From inequalities (1) – (3) it follows that knowing two distances, notably $\rho(y, x^*)$ and $\rho(x^*, x_i)$, it is not hard to obtain low and upper distance bounds

$$\rho(x_i, x^*) - \rho(y, x^*) | \le \rho(y, x_i) \le \rho(y, x^*) + \rho(x_i, x^*) \tag{5}$$

Thus the implication $\forall y \in \mathrm{U},\ \forall x_i, x^* \in \mathrm{X} : \rho(x_i, x^*) \ge 2\rho(y, x^*) \Rightarrow \rho(y, x_i) \ge \rho(y, x^*)$ is true what can be used in the $(k)$- search. Let exact value of distance $\rho(x^*, x_i)$ be unknown and it can be evaluated as

$$\varepsilon_{min} \le \rho(x^*, x_i) \le \varepsilon_{max} \tag{6}$$

Then if for objects $x^*$, $x_i$ the inequality (6) is fulfilled the evaluation of low and upper distance bounds

$$max\{\rho(y, x^*) - \varepsilon_{max}, \varepsilon_{min} - \rho(y, x^*), 0\} \le \rho(y, x_i) \le \rho(y, x^*) + \varepsilon_{max} \tag{7}$$

is true. Indeed, according to the triangular inequality rule and taking into consideration (6) we have

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

then

$$\rho(y, x^*) - \varepsilon_{max} \le \rho(y, x_i) \tag{8}$$

On the other hand, for object s $x_i$ and $x^*$ it is true that $\varepsilon_{min} \le \rho(x_i, x^*) \le \rho(y, x^*) + \rho(y, x_i)$ from where

$$\varepsilon_{min} - \rho(y, x^*) \le \rho(y, x_i) \tag{9}$$

Inequalities (8) and (9) are the low bounds evaluations $\rho(x, x_i)$, and for narrowing the inequality conditions we chose the maximal value. It also should be considered that both evaluations can simultaneously become negative

what is reflected in formula (7). Finally, evaluation of the upper bound $\rho(y, x_i)$ directly follows from the triangular inequality (2) and condition $\rho(x^*, x_i) \le \varepsilon_{max}$.

Let an object $x_i$ be situated 'closer' to the pivot $x_1^*$ than to $x_2^*$, i.e.

$$\rho(x_i, x_1^*) \le \rho(x_i, x_2^*) \tag{10}$$

Then the following inequality takes place

$$max\{(\rho(y, x_1^*) - \rho(y, x_2^*))/2,\ 0\} \le \rho(y, x_i) \tag{11}$$

Indeed, from the triangular inequality we have $\rho(y, x_1^*) \le \rho(y, x_i) + \rho(x_i, x_1^*)$, hence $\rho(y, x_1^*) - \rho(y, x_i) \le$ $\le \rho(x_i, x_1^*)$. Then $\rho(x_i, x_2^*) \le \rho(y, x_2^*) + \rho(y, x_i)$ is hold. Using condition (10), from the inequalities above we get $\rho(y, x_1^*) - \rho(y, x_i) \le \rho(y, x_2^*) + \rho(y, x_i)$ what with account of the expression $\rho(y, x_1^*) - \rho(y, x_2^*)/2$ possible negativity gives relationship (11).

## Metric search models

Let us consider some approaches for $(\delta)$- search support, which make premises for creating effective indexing system for reducing calculation operations on the search stage.

Suppose the distance matrix $d(\mathrm{X})$ is calculated on the pre-processing stage. We shall denote set $\mathrm{X}$ as $\mathrm{X}_0$ and its cardinality $card(\mathrm{X}_0) = n_0$. On the initial search stage we calculate the distance $\rho(y, x^{(0)})$ between the searched object $y$ and some object $x^{(0)} \in \mathrm{X}_0$ which is chosen arbitrary or using some criterion. If $\rho(y, x^{(0)}) \le \delta$ we add object $x^{(0)}$ to a resulting set $\mathrm{Y}$. From inequality (5) it directly follows that the distance $\rho(y, x_j^{(0)})$, $j = \overline{1, n_0}$, $x_j^{(0)} \ne x^{(0)}$ does not satisfy the search criterion $\delta$ if $|\rho(x_j^{(0)}, x^{(0)}) - \rho(y, x^{(0)})| > \delta$. Applying this criterion to all elements $\mathrm{X}_0$ we get a set $\mathrm{X}_1 = \{x_j^{(0)} \in \mathrm{X}_0 : |\rho(x_j^{(0)}, x^{(0)}) - \rho(y, x_j^{(0)})| \le \delta\} \subseteq \mathrm{X}_0$. If $n_0 > card(\mathrm{X}_1) = n_1$ we shall chose by analogy element $x^{(1)} \in \mathrm{X}_1$ and repeat the procedure of evaluating $\rho(y, x_j^{(1)}) \le \delta$ and distance filtration for all $x_j^{(1)} \in \mathrm{X}_1$, $x_j^{(1)} \ne x^{(1)}$ getting in result $\mathrm{X}_2 \subseteq \mathrm{X}_1$, $card(\mathrm{X}_2) = n_2$. The procedure is carried out recursively till step $l$ when $n_{l-1} - 1 = card(X_l)$, $X_l \subseteq X_{l-1}$. In this case distances $\rho(x_j^{(l)}, y)$, $j = \overline{1, n_l}$ are calculated and evaluated directly. Thus the matches number will be equal to $N(\delta) = l + n_l$ where $n_l = card(\mathrm{X}_l)$.

In practice storing distance matrix $d(\mathrm{X})$ 'in whole' is insufficient due to considerable preprocessing time and especially quadratic memory space requirements. One of the ways to solve this problem is to use its 'sparse' form where for some limited set of paired indices $\{k, l\}$, $0 \le k, l \le n, k \ne l$ value $\rho(x_k, x_l)$ is calculated on the pre-processing stage. The natural demand of the methods choosing a set of given combinations is a compromise between storage expenses and number of distance calculating operations on the search stage, which should tend to $N(\delta) = l + n_l$. At the same time one should note that value $N(\delta)$ is a random one in a sense of being dependent on objects space configuration, pivots $x^{(1)}, x^{(2)}, \ldots, x^{(l)}$ choice order and location of the query object

$y$. For instance $N(\delta)$ can be decreased if at first the point $x^{(2)}$ and then the point $x^{(1)}$ are chosen. Thus, indexing methods using the sparse form $d(\mathrm{X})$ (and, therefore operating is limited by information volume), theoretically can perform less matching operations than methods on 'complete' distance matrix $d(\mathrm{X})$.

We shall introduce a set of pivots $\mathrm{X}^* = \{x_1^*, x_2^*, ..., x_k^*\}$. From (5) it follows that low distance value is $\rho(y, x_i) \geq \rho_{\mathrm{X}^*}(y, x_i)$ where $\rho_{\mathrm{X}^*}(y, x_i) = \max\limits_{x^* \in \mathrm{X}^*} |\rho(y, x^*) - \rho(x^*, x_i)|$. This is the simplest filtering method for the sparse distance matrix, where $d(\mathrm{X})$ after corresponding index rearrangement of indexes takes form

$$d(\mathrm{X})_{\mathrm{k}}^* = \begin{pmatrix} 0 & 0 & 0 & ... & \rho(x_1, x_{k+1}) & ... & \rho(x_1, x_n) \\ & 0 & 0 & ... & \rho(x_2, x_{k+1}) & ... & \rho(x_2, x_n) \\ & & 0 & ... & ... & ... & ... \\ & & & 0 & \rho(x_k, x_{k+1}) & ... & \rho(x_k, x_n) \end{pmatrix} = \begin{pmatrix} \rho(x_1, x_{k+1}) & ... & \rho(x_1, x_n) \\ \rho(x_2, x_{k+1}) & ... & \rho(x_2, x_n) \\ ... & ... & ... \\ \rho(x_k, x_{k+1}) & ... & \rho(x_k, x_n) \end{pmatrix} \quad (12)$$

We will emphasize that we do not store distances between pivots in $d(\mathrm{X})_{\mathrm{k}}^*$ since $\rho(y, x_j^*), j = \overline{1,k}$ are calculated directly during the search. It also should be noted that the introduced approach can be interpreted as a mapping $(\mathrm{X}, \rho) \to (\mathrm{R}^k, L_\infty)$ and search in $k$ -dimensional space.

Another way of creating index structure without calculating and storing $d(\mathrm{X})$ 'in whole' is to analyze the structure of the data set on the base of distances between objects and then create a partition (possibly nested) of $\mathrm{X} = \{\mathrm{X}^{(j)}\}$, $j = \overline{1,m}$, where

$$\left.\begin{array}{r} \forall j, j' \in \{1, \ldots, m\} : j \neq j' \Rightarrow \mathrm{X}^{(j)} \bigcap \mathrm{X}^{(j')} = \varnothing, \\ \mathrm{X}^{(1)} \bigcup \ldots \bigcup \mathrm{X}^{(m)} = \mathrm{X}, \end{array}\right\} \quad (13)$$

is fulfilled. Here the equivalence relation built on the base of function $\rho$ can be exploited on the search stage: we do not consider those sets $\mathrm{X}^{(j)}$ which element $x_i^{(j)} \in \mathrm{X}^{(j)}$ is not equivalent to the query object $y$. Here important role plays determination of low and upper bounds (7), (12) $\rho(y, x_i)$ which allow to estimate the distance from $y$ to the elements of $\mathrm{X}^{(j)}$, $j = \overline{1,m}$, separately or using distances to other sets $\mathrm{X}^{(j')}$, $j' = \overline{1,k}$, $j' \neq j$.

Let us consider another way of partitioning $\mathrm{X}$. We shall choose pivot $x_j^*$ which has index $j$ in matrix $d(\mathrm{X})$, and determine the distance to all the rest of the objects $d(\mathrm{X})_{j,1}, d(\mathrm{X})_{j,2}, \ldots, d(\mathrm{X})_{j,n}$. We shall sort values of raw $j$ in ascending order reassign indices $d^*(\mathrm{X})_{j,1}, d^*(\mathrm{X})_{j,2}, \ldots, d^*(\mathrm{X})_{j,n}$ and define the distance to the median object $\rho(x_j^*, d^*(\mathrm{X})_{j,k}) = \mathrm{M}$, $k = \lceil n/2 \rceil$. We shall introduce partition of $\mathrm{X}$ into two equivalence classes $\mathrm{X}_{\leq}$ and $\mathrm{X}_{>}$ where $\mathrm{X}_{\leq} = \{x_i \in \mathrm{X} : \rho(x_j^*, x_i) \leq \mathrm{M}\}$, $\mathrm{X}_{>} = \{x_i \in \mathrm{X} : \rho(x_j^*, x_i) > \mathrm{M}\}$. In this case on the search stage under $\rho(x_j^*, y) \leq \mathrm{M} - \delta$ it is necessary to search only class $X_{\leq}$ and under $\rho(x_j^*, y) > \mathrm{M} + \delta$ only class $\mathrm{X}_{>}$. Thus producing such a partition can allow us excluding from the consideration half of the set elements. But in the worse case when $\mathrm{M} - \delta < \rho(x_j^*, y) \leq \mathrm{M} + \delta$ is true, search algorithm has to consider both

branches $\mathrm{X}_{\leq}$ and $\mathrm{X}_{>}$. It should be emphasized that this method is to be used recursively.

We shall introduce into consideration equivalence relation based on the closeness to the pivot. As before let us chose set $\mathrm{X}^* = \{x_1^*, x_2^*, ..., x_k^*\}$. Then elements $x_j^*$ produce partition $\mathrm{X} = \{\mathrm{X}_s^*\}$, $s = \overline{1,k}$ such that

$$\mathrm{X}_s^* = \{x_i \in \mathrm{X} : \forall t = \overline{1,k}, t \neq s \; \rho(x_i, x_s^*) < \rho(x_i, x_t^*)\} \tag{14}$$

Such criteria coincide with definition of Voronoi cell in Euclidean space. Partition introduced this way allows to use evaluation (11) of the low distance bound. Indeed, the criterion $\rho(x_i, x_s^*) \leq \rho(x_i, x_t^*)$ for $x_i \in \mathrm{X}_s^*$, $t \neq s$ is fulfilled by (14). Then from (11) for $k$ pivots it follows that the low bound of $\rho(y, x_i)$ for $x_i \in \mathrm{X}_s^*$ will be

$$\rho_{min}^{(1)}(s) = max\Big\{ \max_{t=\overline{1,k}(t\neq s)} \{(\rho(y, x_s^*) - \rho(y, x_t^*)/2)\}, 0\Big\} \tag{15}$$

Let $\varepsilon_{max}(s) = \max\limits_{x_i \in \mathrm{X}_t^*} \rho(x_s^*, x_i)$ be a cover radius for the partition $\mathrm{X}_s^*$. Then according to (7) the evaluation low distance bound $\rho_{min}^{(2)}(s) = \rho(y, x_s^*) - \varepsilon_{max}(s)$ is true. Let minimal and maximal distance evaluations between partitions be calculated on the preprocessing stage for $s, t = \overline{1,k}$.

$$\varepsilon_{min}(s,t) = \min_{x_i \in \mathrm{X}_t^*} \rho(x_s^*, x_i), \; \varepsilon_{max}(s,t) = \max_{x_i \in \mathrm{X}_t^*} \rho(x_s^*, x_i)$$

Then $\varepsilon_{min}(s,t)$ and $\varepsilon_{max}(s,t)$ under $s \neq t$ are evaluations of $\varepsilon_{min}$ and $\varepsilon_{max}$ distance $\rho(x_s^*, x_t^*)$ in (7). For $s = t$ $\varepsilon_{max}(s,s) = \max\limits_{x_i \in \mathrm{X}_s^*} \rho(x_s^*, x_i) = \varepsilon_{max}(s)$. Hence it is legitimate to claim that evaluation $\rho_{min}^{(2)}(s)$ is a special case of evaluation $\rho_{min}^{(3)}(s) = max\{\max\limits_{t=\overline{1,k}} \{\rho_{min}^{(3)}(s,t)\}, 0\}$ where

$$\rho_{min}^{(3)}(s,t) = max\{\rho(y, x_s^*) - \varepsilon_{max}(s,t), \varepsilon_{min}(s,t) - \rho(y, x_s^*)\} \tag{16}$$

The final maximal low bound of distance $\rho(y, x_i)$, $x_i \in \mathrm{X}_s^*$ is defined as $\rho_{min}(s) = max\{\rho_{min}^{(1)}(s), \rho_{min}^{(3)}(s)\}$.

To make search algorithm on partition index structure more optimal, we propose to carry out the following steps during the search. Let $\mathrm{E}$ be a set of not processed pivots $x_s^*$ which form corresponding regions $\mathrm{X}_s^*$, $s = \overline{1,n}$, and $\mathrm{T}$ be a set of regions $\mathrm{X}_s^*$ which cannot be dropped by partition index structure. Then iteratively get the first pivot $x_s^* \in \mathrm{E}$, calculate $\rho(x_s^*, y)$, estimate the low bound of all $\rho(x_t^*, y)$, $x_t^* \in \mathrm{E}$ using (16) and remove from $\mathrm{E}$ those $x_t^* : \rho_{min}^3(s,t) > \delta$ or put corresponding to $x_t^*$ set $\mathrm{X}_t^*$ to $\mathrm{T}$. Also after each iteration remove considered pivot $x_s^*$ from $\mathrm{E}$, thus iterative procedure stops when some pivots $x_s^*, s = \overline{1,t}$ are eliminated by (16) and distance to the rest of pivots is calculated. After it we argue to apply (15) criteria for all pairs $x_s^*, x_t^* \in \mathrm{E}$, $s \neq t$ since a range $(\varepsilon_{min}(s,t), \varepsilon_{max}(s,t))$ could be large and therefore could produce large low bound of $\rho(y, x_i)$, $x_i \in \mathrm{X}_t^*$ or $x_i \in \mathrm{X}_s^*$.

## Results of experiments and conclusion

A number of tests have been performed on set of points in $\mathrm{R}^2$ space with $L_2$ metric. We used two configurations of data distribution: uniform and with formed clusters. We implemented the following index structures to preprocess the initial data set: i) indexation on full matrix (1); ii) indexation on sparse matrix (12); iii) indexation via binary tree with branches $\mathrm{X}_{\leq}$ and $\mathrm{X}_{>}$; iv) indexation via compact partition using criteria (14) and iterative procedure which exploits (15) and (10) low bounds.

The purpose was to calculate the matches' number during the search on uniform and clusters distribution of objects and its dependency on the query object position and data set configuration.

Below the results of the tests with the following experiment parameters are presented. The data for uniform distribution consisted of 1024 points, for the clusters one there were 16 clusters, with cluster cardinality mean equal to 64 and variance equal to 10 of (1024 elements in total). Both sets of points coordinated were within the range [0;256] (data square here and after). Variance of single cluster points location was set to 6 for both coordinates. It was allowed that clusters could overlap. We used $k = \sqrt{n}$ parameters for indexation ii), and created one-level partition with $k = \sqrt{n}$ number of pivots in indexation iv).

First experiment examined dependency of the index structure on the position of the query object. We generated uniform and cluster data structures, created all index structures and randomly chose query object from the data square 500 times, tracking the number of matches of each data structures for all queries. We then calculated the mean and variance of matches count for each index structure (Table 1).

Table 1. Dependency of the matches' number on different data configurations

| Data configuration | Full matrix | | Sparse matrix | | Partition search | | Binary search | |
|---|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| *uniform* | 23.56 | 4.82 | 50.98 | 4.58 | 107.78 | 36.18 | 90.08 | 23.7 |
| *clusters* | 30.63 | 29.73 | 55.19 | 26.27 | 71.2 | 71.39 | 74.362 | 48.49 |

As it was expected indexation which exploits a full distance matrix i) outperforms the other ones while indexation on sparse matrix ii) keeps the second place. Indexations iii) and iv) have approximately equal efficiency. We can claim that when objects of database tend to form clusters the number of matches does not vary dramatically for all indexing methods, they only differ notably on variance and therefore in some cases performance of methods iii) and iv) can take much more time than it was expected. On the other hand, when distribution of objects is uniform then indexation i) perform 4 times better results than iii) and iv) while indexation ii) performs 2 times better results.

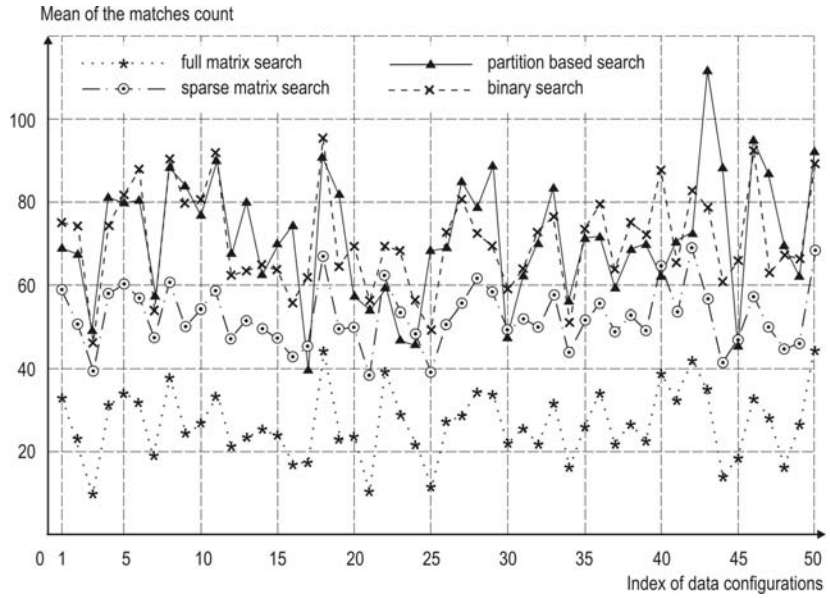


Figure 1. Results of matches count dependency on change of cluster data configuration

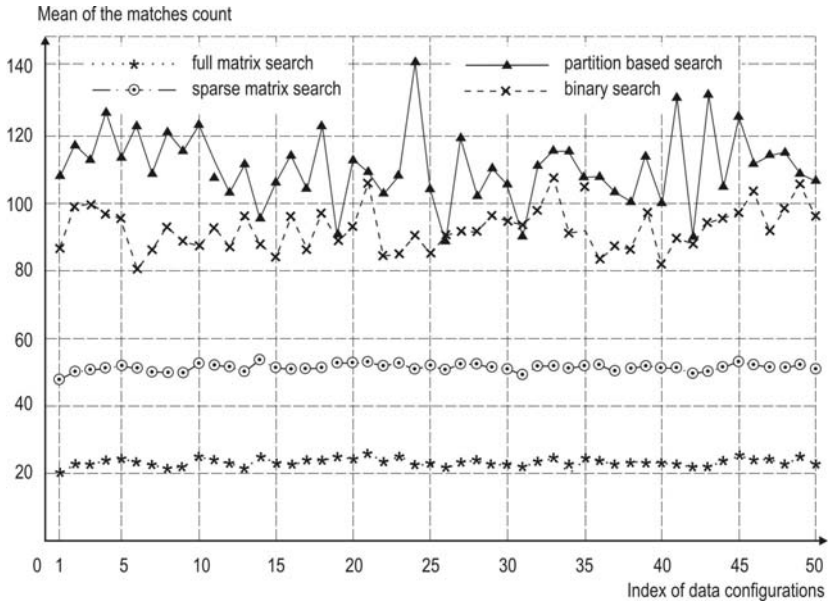


Figure 2. Results of matches count dependency on change of uniform data configuration

In the second experiment we tried to track dependency of index structures efficiency on different data configurations. We created 50 random configurations of uniform and cluster data distributions and ran routines of the first experiment changing query object position 20 times. We found out that statistics of the indexations efficiency was almost the same compared to first experiment results, only variance grew a bit. On Figures 1, 2 an average number of distance computations on 50 iterations for indexation algorithms i) – iv) is given for uniform and cluster data configuration. From here we can conclude that only index structures i) and ii) can guarantee almost/rather constant low bound of matches count on uniformly distributed data set solely.

## Conclusion

We have proposed a novel indexing structure using sparse distance matrices for the image search with queries 'ad exemplum' which considering embedded partitions of the images. The experimental exploration of the method has proved it to be fast and efficient. The future work will be directed for investigation of the pivots choice method and possibility to use clustering methods for distance matrices analysis which would provide effective using of the partition metric for content image retrieval.

## Bibliography

[Greenspan et al., 2004] H. Greenspan, G. Dvir, Y. Rubner. Context-Dependent Segmentation and Matching in Image Database. Computer Vision and Image Understanding. Vol. 93, No. 1, 2004, pp. 86-109.

[Yokoyama, Watanabe, 2007] T. Yokoyama, T. Watanabe. DR Image and Fractal Correlogram: A New Image Feature Representation Based on Fractal Codes and Its Application to Image Retrieval. In: Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag. Vol. 4351, 2007, pp. 428-439.

[Rubner et al., 2000] Y. Rubner, C. Tomasi, L.J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision. Vol. 40, No 2, 2000, pp. 99-121.

[Cheng et al., 2005] W. Cheng, D. Xu, Y. Jiang, C. Lang. Information Theoretic Metrics in Shot Boundary Detection. In: Knowledge-Based Intelligent Information and Engineering Systems. Khosla R., et al. (eds.). Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg. Vol. 3683, 2005, pp. 388-394.

[Wang et al., 2005] D. Wang, X. Ma, Y. Kim. Learning Pseudo Metric for Intelligent Multimedia Data Classification and Retrieval. Journal of Intelligent Manufacturing. Vol. 16, No 6, 2005, pp. 575–586.

[Kinoshenko et al., 2007] D. Kinoshenko, V. Mashtalir, V. Shlyakhov A Partition Metric for Clustering Features Analysis // International Journal 'Information Theories and Applications'. Vol. 14, No 3, 2007, pp. 230-236.

[Bohm et al., 2001] C., Berchtold S., Keim D. Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases. ACM Computing Surveys. Vol. 33, No. 3, 2001, pp. 322–373.

[Chavez et al., 2001] Chavez E., Navarro G., Baeza-Yates R., Marroquin J.L. Searching in Metric Spaces. ACM Computing Surveys. Vol. 3, No. 3, 2001, pp. 273–321.

[Hjaltason, Samet, 2003] Hjaltason G., Samet H. Index-driven Similarity Search in Metric Spaces. ACM Transactions on Database Systems. Vol. 8, No. 4, 2003, pp. 517-580.

## Authors' Information

*Dmitry Kinoshenko – Researcher; Kharkov National University of Radio Electronics, Lenin ave. 14, Kharkov 61166, Ukraine; e-mail: Kinoshenko@kture.kharkov.ua*

*Vladimir Mashtalir – Professor, Dean of Computer Science faculty; Kharkov National University of Radio Electronics, Lenin ave. 14, Kharkov 61166, Ukraine; e-mail: Mashtalir@kture.kharkov.ua*

*Elena Yegorova – Senior researcher; Kharkov National University of Radio Electronics, Lenin ave. 14, Kharkov 61166, Ukraine; e-mail: Yegorova@kture.kharkov.ua*