Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 40 – 49

Procedia Computer Science

2013 International Conference on Computational Science

Interactive data mining by using multidimensional scaling

Piotr Pawliczek[a,b], Witold Dzwinel*[b]

[a]University of Texas, Department of Biochemistry and Molecular Biology, Houston, TX 77030, USA
[b]AGH University of Science and Technology, Department of Computer Science, Al.Mickiewicza 30, 30-059 Kraków, Poland

Abstract

Blind choice and parameterization of data mining tools often yield vague or completely misleading results. Interactive visualization enables not only extensive exploration of data but also better matching of clustering/classification schemes to the type of data being analyzed. The multidimensional scaling (MDS), which employs particle dynamics to the error function minimization, is a good candidate to be a computational engine for interactive data mining. However, the main disadvantage of MDS is both its memory and time complexity. We developed novel SUBSET algorithm of a lower complexity, which is competitive to the best, currently used, MDS algorithms in terms of efficiency and accuracy. SUBSET employs reduced dissimilarity matrix, which structure allows for efficient usage of both multi-core CPU and SIMD GPU processor architectures. Consequently, SUBSET enables visualization of datasets consisting of an order of 10^5 data items on a standard personal computer or laptop. We compare a few strategies of dissimilarity matrix reduction and we present typical timings obtained by respective MDS algorithms on selected multithread CPU and GPU architectures.

Keywords: multidimensional scaling; method of particles; incomplete distance matrix; multicore CPU; GPU.

1. Introduction

Data mining of large datasets consisting of data items $\boldsymbol{O}_i \in \boldsymbol{\Omega}$, ($i$=1,…,$M$), where $\boldsymbol{\Omega}$ is an abstract data space, involves application of many machine learning tools such as classifiers, regression and clustering schemes. Many of them are specialized for analysis of a special case of $\boldsymbol{\Omega}$ represented by $\mathbf{Y}$ space of multidimensional

* Corresponding author. tel.: +48 662130188, fax: +48 12 617 51 72
E-mail address: dzwinel@agh.edu.pl

doi:10.1016/j.procs.2013.05.167

feature vectors $\mathbf{y}_i$, $\mathbf{Y}=\{\mathbf{y}_i=(y_{i1},..y_{iN})\}_{i=1,...,M}$, where $N=\dim\mathbf{Y}$ (see Fig.1a). However, in general, the data items $\boldsymbol{O}_i$ can have more sophisticated structures, which cannot be directly represented by the feature vectors. This problem can be partially overcome provided that it is possible to define a dissimilarity measure $\delta(\boldsymbol{O}_i,\boldsymbol{O}_j)\rightarrow\Re^1$ between data items $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$. We assume that $\delta(.,.)$ is symmetric and $\delta(\boldsymbol{O}_i,\boldsymbol{O}_i)=0$. Particularly, $\delta(,)$ can be a distance obeying also the condition of triangle inequality. In general, the $\boldsymbol{\Omega}$ space topology is represented by dissimilarity matrix $\boldsymbol{\Delta}=\{\delta_{ij}\}_{MxM}$. The vector representation of $\boldsymbol{\Omega}$ can be derived by employing multidimensional scaling (MDS) procedure [1-3].

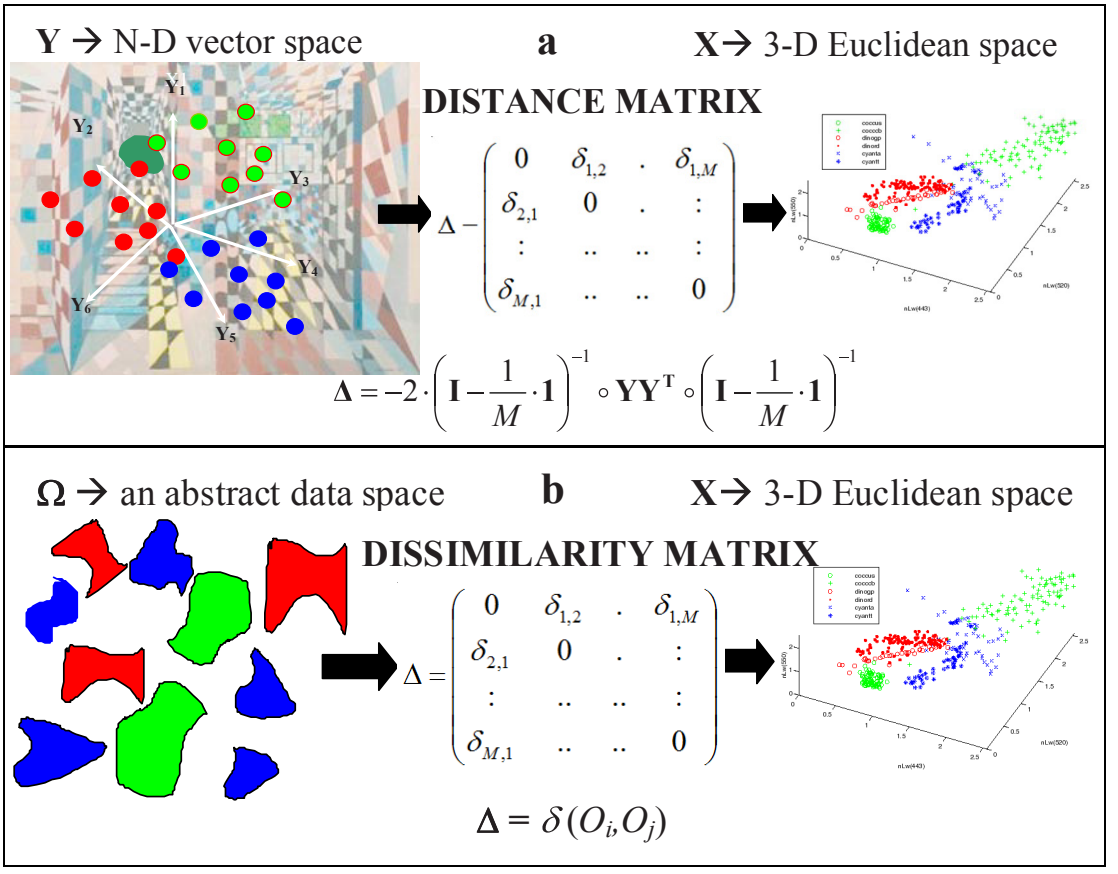

Fig.1. Multidimensional scaling applied for visualization in 3-D Euclidean space **X** data from a) a multidimensional feature space **Y,** b) an abstract $\boldsymbol{\Omega}$ space for which only dissimilarity matrix $\boldsymbol{\Delta}$ is known (e.g., dissimilarity measure between shapes).

Multidimensional scaling (MDS) (see e.g. [1,2,3]) is a bijection $B:\boldsymbol{\Omega}\rightarrow\mathbf{X}$ of a "source" space of abstract items $\boldsymbol{\Omega}=\{\boldsymbol{O}_i;\ i=1,...,M\}$ onto a "target" vector space $\Re^n\ni\mathbf{X}=\{\mathbf{x}_i=(x_{i1},..x_{in})\}_{i=1,...,M}$, where $n=\dim\mathbf{X}$, which reproduce topological structure of $\boldsymbol{\Omega}$ in **X** in respect to a given error criterion (see Fig.1b) .

Let us define the Euclidean matrix $\mathbf{d}=\{d_{ij}\}_{MxM}$ in the target vector space **X,** where d_{ij} is the Euclidean distance between vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ which correspond to $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$, respectively. We assume that to preserve topological structure of $\boldsymbol{\Omega}$ in **X** an overall error $F(\|\boldsymbol{\Delta}-\mathbf{d}\|)$ should be minimized, where $F(.)$ - called the "stress" function [1-3] - is an increasing function $F:\Re^1\rightarrow\Re^1$. The value of $F(.)$ represents a discrepancy measure between dissimilarities $\boldsymbol{\Delta}$ from $\boldsymbol{\Omega}$ and corresponding distances **d** from **X**. The resulting matrix $\mathbf{X}=(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots,\mathbf{x}_M)$, which minimizes the "stress" function $F(.)$, is the final outcome of multidimensional scaling. This way

each data item $\boldsymbol{O}_i$ can be represented in $\mathbf{X}$ by a corresponding feature vector $\mathbf{x}_i$. Having vector representations $\mathbf{x}_i$ of data items $\boldsymbol{O}_i$, one can employ classical machine learning algorithms for data mining without using complicated syntactic rules and languages.

However, it is well known that the quality of a particular classification or clustering scheme depends strongly on data analyzed. Moreover, its parameters should also be optimized in the context of a specific data type. Blind selection and parameterization of classification/clustering schemes too often yield vague or completely misleading results. Therefore, much time has to be spent adapting the machine learning algorithms to data being processed. Interactive data visualization allowing for direct manipulation on data sets and instant observation of its effects of (i.e., interactive data mining), can radically improve data understanding. It enables for finding the best match of classification/clustering procedures to data being explored and deeper penetration of their parameter space.

As shown in [4], multidimensional scaling is a good candidate to be a methodological framework of interactive visualization. Because it allows for reconstructing the topology of an original data space defined by the dissimilarity matrix $\boldsymbol{\Delta}$ in a target vector space $\mathbf{X}$, we can use it particularly in dimensions which can be explored visually, i.e., dim $\mathbf{X}$=3 or 2 (see Fig.1). In general, the distance matrix $\boldsymbol{\Delta}=\{\delta_{ij}\}_{MxM}$ can be non-Euclidean matrix of dissimilarities in an abstract target space, while $\mathbf{d}=\{d_{ij}\}_{MxM}$ is its corresponding Euclidean matrix in a low dimensional (3(2)-D) target space (see Fig.1b).

In [5,6] we showed that:

1. by representing every data item O_i by a corresponding point particle $\mathbf{x}_i$ in 3(2)-D target space;
2. by assuming that every two particles i and j interact with each other via semi-harmonic forces dependent on the discrepancy $||\delta_{ij}-d_{ij}||$ between corresponding particle distances in the source and the target spaces;
3. by simulating Newtonian time evolution of initially random configuration of particles in a dissipative environment;

the particle system converges to a stationary state with a minimal potential energy $E(\mathbf{X})$

$$E(\mathbf{X}) = \min_{\mathbf{X}} F\left(\left\|\boldsymbol{\Delta}-\mathbf{d}\right\|\right) = \min_{\mathbf{X}} \sum_{i<j} w_{ij}\left(\delta_{ij}^{k} - d_{ij}^{k}\right)^{m} . \tag{1}$$

The final particle configuration $\mathbf{X}=\{\mathbf{x}_i\}_{i=1...M}$ represents the result of MDS mapping into 3(2)-D target space. The values of k,m are the parameters of the error function $F(.)$ (e.g. k=1 and m=2), which represents discrepancy error between dissimilarities $\boldsymbol{\Delta}$ in the source space and corresponding Euclidean distances $\mathbf{d}$ in the target space. Assuming very slow dissipation of kinetic energy of the full particle system, the global minimum (or, at least, a "good" local minimum close to the global one) of $F(.)$ will be reached. This minimization procedure fits perfectly for interactive visualization purposes mimicking self-adaptation process in course of simulation (see [4,5]). By changing the interaction potential (i.e., the type of discrepancy measure), interactive control of simulation parameters and by removing, inserting and stopping selected particles, one can easily penetrate data topology both controlling its relaxation dynamics and exploring interactively the final particle configuration.

However, this robust heuristics - similarly to other MDS algorithms which employ full distance matrix for data representation - suffers $O(M^2)$ memory and time complexity. This fact disables interactive visualization of larger data sets consisting of $M>10^5$ items on up-to-the-date personal computers. As shown in [7,8] this situation can be improved by using supercomputers and HP clusters. Nevertheless, this computational problem remains too demanding for smaller computer systems.

In general, to find analytically the minimum of criterion (1) the system of $n \times M$ nonlinear equations in n-dimensional target space should be solved. However, such the system is strongly overdetermined [6]. It means that just a subset S of all distances from $\boldsymbol{\Delta}$ is sufficient to clearly define data topology. The proper choice of S with minimal number of distances, which preserve the topological structure of data, could considerable improve the computational efficiency of multidimensional scaling.

In this paper we demonstrate that by using only a small fraction of distances from $\boldsymbol{\Delta}$ and simultaneously preserving high coherence of data, we are able to increase the efficiency of MDS by orders of magnitude with a

small deterioration of the final result. In the following sections we present the result of integration of three factors which decide about MDS efficiency, i.e., robust heuristics for error minimization (i.e., particle method [4]), usage of incomplete dissimilarity matrix and efficient parallel algorithms. We compare typical timings of tests performed on multi-core CPU and GPU processors for a few strategies of valid distances selection.

2. MDS with reduced dissimilarity matrix

Let us assume that the feature vectors $\mathbf{x}_i$ ($i=1,...,M$), represented by the points in N-D space, are connected by rigid bonds of the length of dissimilarity values between corresponding data items and the whole data set is fixed (cannot rotate and move) in $\mathbf{\Omega}$. The minimal number of bonds which unambiguously defines such the system is approximately equal to:

$$L_{Min} = N \cdot M - N \cdot (N+1)/2 \quad \text{and} \quad \mu = L_{Min}/L_{Max} = \frac{2N}{(M+1)} \cdot \left(1 - \frac{(N+1)}{2M}\right), \tag{2}$$

assuming that $M>N+1$ and $L_{Max}=M \cdot (M-1)/2$. Then for $M=cN$

$$\mu \rightarrow 1/c, \tag{3}$$

i.e., larger value of c ($M>>N$) means that smaller fraction of distances decides about topological structure of data. Therefore, the simplest method of reduction of $O(M^2)$ computational complexity is to calculate only a subset $\mathbf{S}=\{D_{ij}, z=\#D_{ij}\}$, of $z<<L_{Max}$ selected dissimilarities from $\mathbf{\Delta}$. According to (2) and (3) we can expect that the number of distances z taken in random from the full set of L_{Max} distances allowing for unambiguous reconstruction data topology can be proportional to L_{Min}. To check this presumption we selected three datasets representing various multidimensional data structures listed in Table 1.

Table 1. The list of datasets used for tests: name, number of feature vectors M, dimensionality N and short description.

Name	#vectors M	N	Description
H(1..,i,…256)	i x 1024 $M \leq 2.68 \times 10^5$	60	These datasets were artificial generated. They consist of two equal clusters. One half of coordinates are randomly selected from interval [-1,1]. Remaining coordinates are randomly selected from interval [-1, 0] or [0, 1].
WDBC	569	30	The WDBC dataset was taken from the UCI Machine Learning Repository ([9]). Nine versions of this dataset were generated by choosing randomly the vectors from the original dataset.
PENDIGITS	3498	16	This is one of the UCI Machine Learning Repository [9] datasets. Each vector corresponds to one handwritten digit. Nine versions of this dataset were generated by choosing the vectors randomly from the original dataset.

The Hi (i=1,…,256) test bed consists of multidimensional feature vectors grouped in two separated clusters while WDBC represents two adhering data clouds. The PENDIGITS data set consists of feature vectors belonging to several non-compact and blurred data structures. The number of feature vectors M in the test beds varied from hundreds to hundred of thousands.

Each test has been performed ten times with various seeds initializing the uniform random number generator. On the basis of ten tests the values of the means and standard deviations of $F(.)$ are controlled for each dataset. Every test run began with a ballot of distances that were used for computations. Thus, the subset $\mathbf{S}$ of z distances is selected. The quality of the results we estimate calculating the values of stress functions (Eq.1) assuming that $w_{ij}=1$; $l=1$; $m=2$, both for the reduced and full distance matrices. The ratio $q(z)$, which is used to estimate the quality of the approximation, is calculated by using the following expression:

$$q(z) = F(\text{reduced dissimilarity matrix})/F(\text{full dissimilarity matrix}) \cdot 100\% \,. \quad (4)$$

It is equal to 100% when all the distances are used in MDS mapping and is larger than 100% for less accurate fit when $z<L_{Max}$. Because, this stress function is not normalized (w_{ij}=1), the relative error component connected with large distances is much smaller than that computed for shorter distances. Typical results obtained for H*i*, WDBC and PENDIGITS datasets are presented in Fig.2. The plots display the minimum number of distances z versus the size of dataset M needed to obtain the required quality of mapping $q(z)$. The assumed values of $q(z)$ are equal to 100%, 103%, 106%, and 109%, respectively.

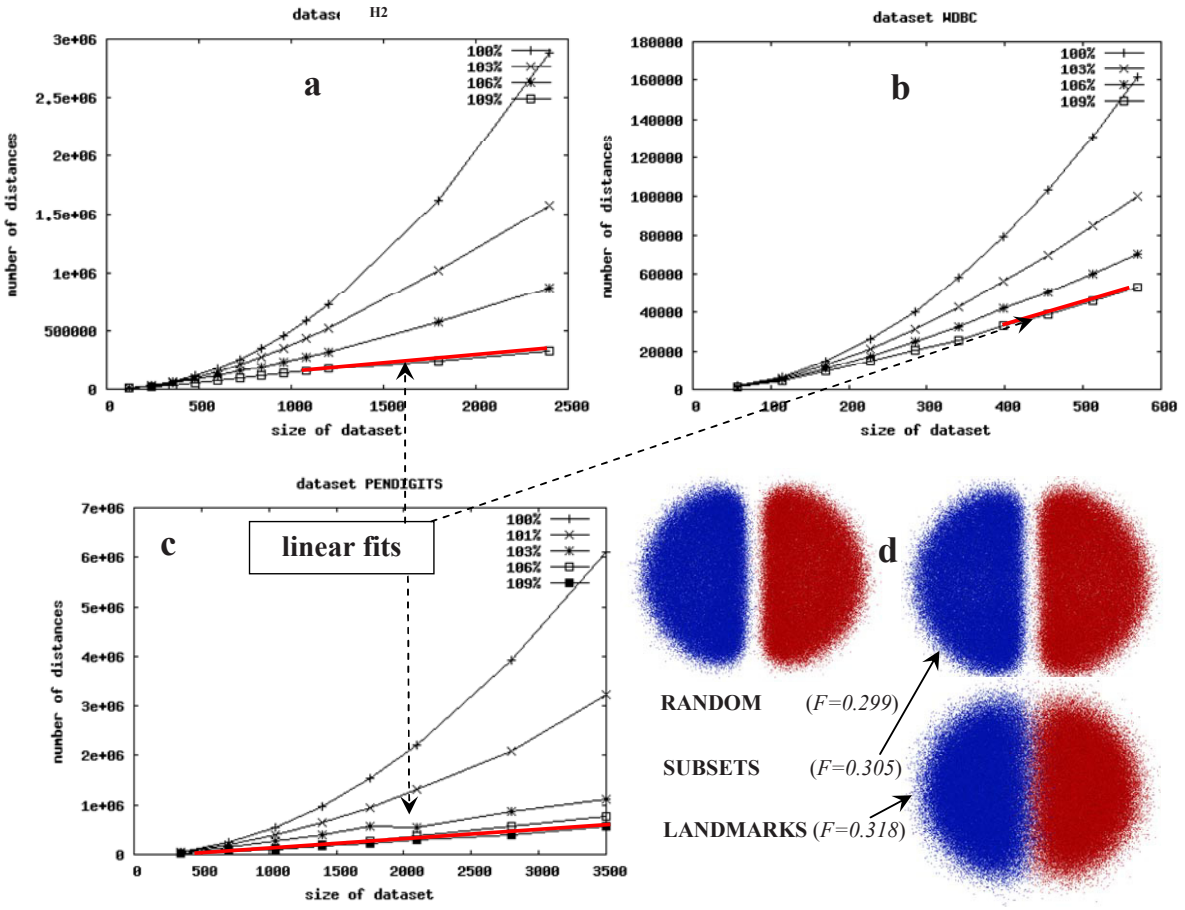

Fig.2. The minimum number of distances z needed to obtain the required quality of mapping $q(z)$ versus M. The plots represent a) H2 b) WDBC and c) PENDIGITS datasets, respectively. In panel d) we display the results of particle based MDS mapping using 3 various choices of distances (only 1% of distances were chosen) for H256 data set (2.68×10^5 feature vectors). The value of stress F is given in parenthesis.

As shown in Fig.2a-c, assuming relatively high quality of mapping (with $F(.)$=109%), z is nearly linear function of M for datasets, where M greatly outnumbers the dimensionality N of the feature space **Y**. This conclusion follows from Eq.3, which shows that for $M>>N$ a lesser fraction of distances μ is required to reconstruct the topologies of source feature spaces. Linear dependence of the number of distances z on M, sufficient for reconstruction of source **Y** space with a given quality $q(z)$, is evident especially for diversified test

bed PENDIGITS and for two other datasets WDBC and H2 for $M>500$ and $M>1000$, respectively. This effect is even stronger for data consisting of many small distant clusters, such as PENDIGITS, and attenuated for topologies represented by two populated clusters (such as H2 and WDBC). This is because the histogram of distances for the former data set is shifted to the long distances while for the latter to the shorter ones. This substantial error component connected with improper reconstruction of shorter distances is greatly reduced for PENDIGITS dataset. Comparing the results of MDS mapping using various ways of z distances selection we see clearly the advantage of a simple random choice.

The final results of embedding of PENDIGITS dataset in 3-D Euclidean space are displayed in Fig.3. For $F(.)=100\%$ (left panel) full dissimilarity matrix was employed by MDS, while for $F(.)=109\%$ (right panel) only 7% of distances were used. At the first sight, it seems that pictures in the two panels are identical. However, being more careful, one can notice that resulting clusters for $q(z)=109\%$ are more blurred.

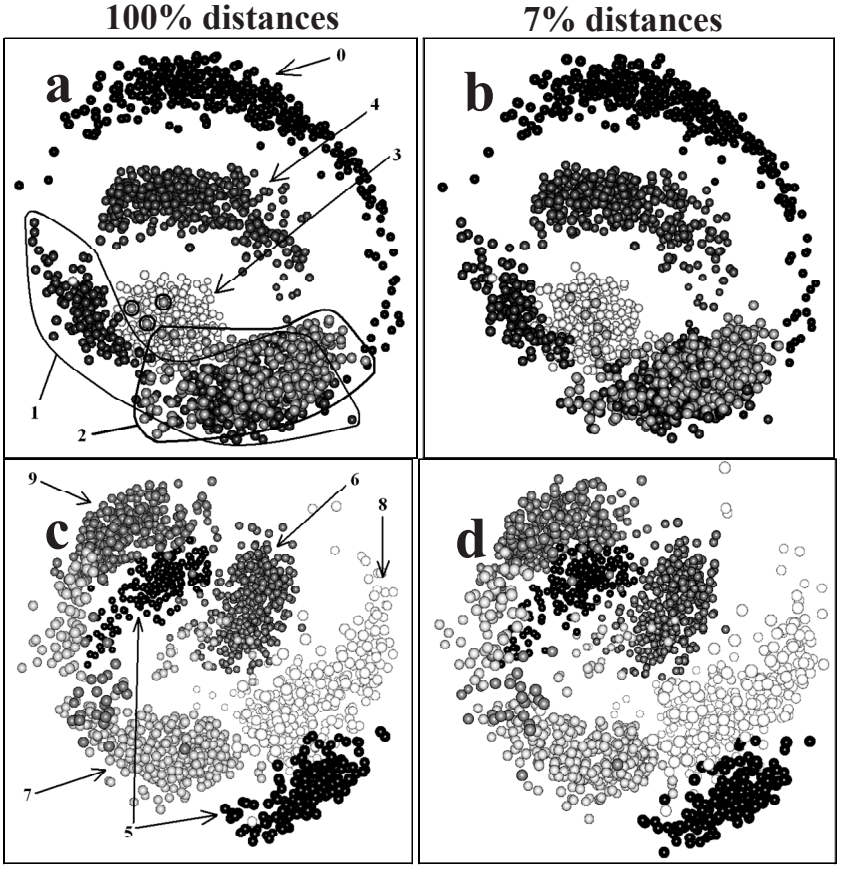

Fig.3 The results of multidimensional scaling of PENDIGITS dataset into 3D space for $q(z)=100\%$ (the left panel); and b) $q(z)=109\%$ (the right panel). The clusters corresponding to digits 0-4 and 5-9 are presented as separate projections a),b) and c),d), respectively.

This effect can be seen even better in Fig.4. The picture of size 75x105 pixels was fragmented onto separate pixels, and scattered randomly in 2-D space. The Euclidean matrix Δ is only information about the proper location of the pixels. As shown in Fig.5, the MDS particle method [4,5] reconstructs properly the picture on the basis of reduced distance matrix. As shown in Fig.4, a fraction of pixel-to-pixel distances is needed to obtain the legible reconstruction.

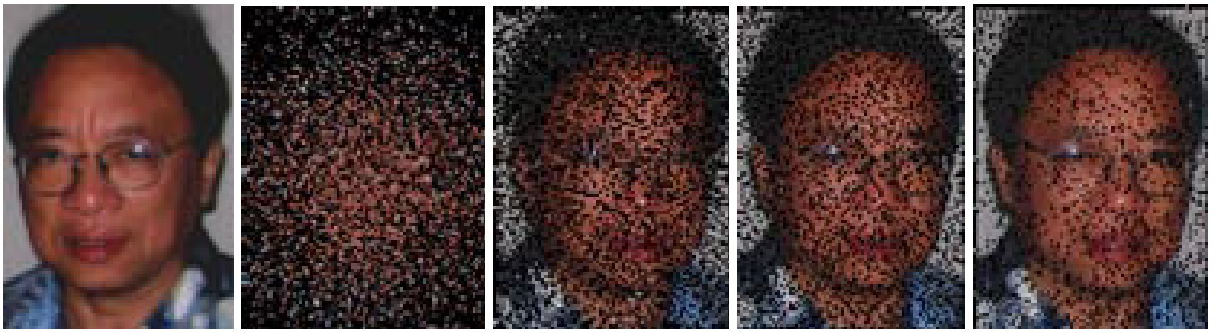

Fig.4. The results of multidimensional scaling of 7875 pixels of a 75x105 picture, initially scattered randomly in 2-D Euclidean space. The number of distances used for the picture reconstruction is given in the parenthesis. The MDS based on virtual particle method was used. The results presented were obtained after 10^4 time-steps.

3. MDS with reduced distance matrix on parallel processors

The RANDOM algorithm presented in the previous section, though efficient and easy, it requires sparse random matrices representation to save CPU memory. The search for matrix elements in such the representation is very inefficient, especially on SIMD architectures requiring data locality such as GPU processors. This problem can be avoided by using more structured distance matrices such as in LANDMARK algorithm [10] (see Fig.5a). In this case all the distances are computed pairwise only between k selected vectors (landmarks), while for the rest of vectors only distances to the landmarks are calculated. The same strategy was applied in the fastest known MDS algorithm GLIMMER [11], which is based on Multigrid MDS [12] and Chalmers [13] algorithms and tuned to the GPU architecture by using low level GPU instructions. However, as we will show here, the result obtained by GLIMMER represents usually a local minimum of discrepancy function (1) instead of the global one.

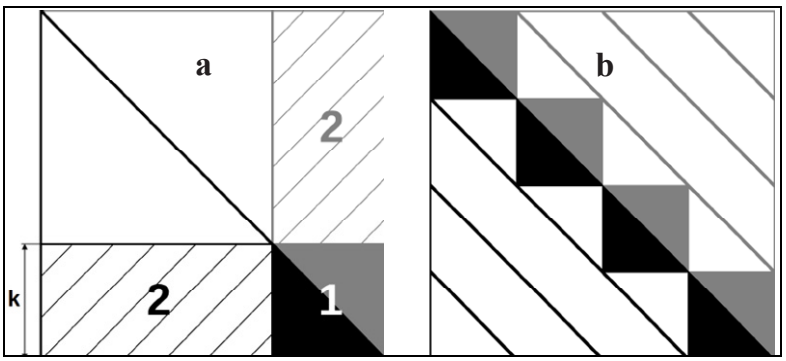

Fig.5. Fragments of distances table used in a) LANDMARK/GLIMMER and b) SUBSET algorithms.

To improve data locality we have developed a novel MDS particle based algorithm SUBSET which uses the distance matrix structure shown in Fig.5b. The input dataset is divided into p almost equal (small differences are allowed) subsets. All the distances inside the subsets are calculated. The data items belonging to each

subset are sorted according to their indices. The distances between each vector from each subset to the respective vectors (same position in the sorted list) from other subsets and their nearest predecessor and successor are computed. For terminal data items (with indices i=1 and i=M), periodic boundary conditions are applied. As shown in Fig.5b, for each of data item apart from all distances to the members of the same subset, additional distances to three data items from other subsets are computed. As shown in Fig.2d, the quality of the results obtained by SUBSET is comparable to that obtained for RANDOM distances setting.

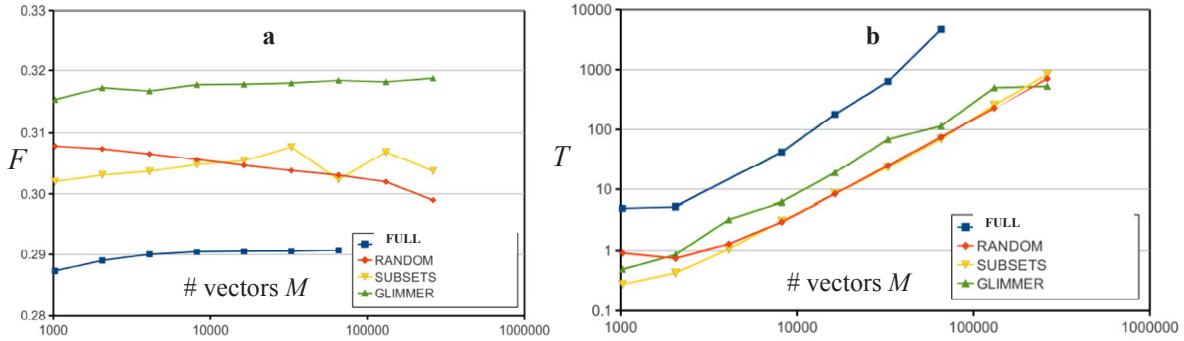

Fig.6. The values of the stress function $F(.)$ (1) (w=1, k=1, m=2) and simulation time T (in seconds) obtained for MDS mapping of H(i=1…256) ($M \rightarrow 2.68 \times 10^5$ vectors) datasets into 3-D target space by using particle based MDS with full distance matrix (FULL), SUBSET and GLIMMER algorithms, respectively. The parallel versions of the algorithms implemented using OpenMP environment and run on 2xIntel Xeon X5670 (12 threads) were compared.

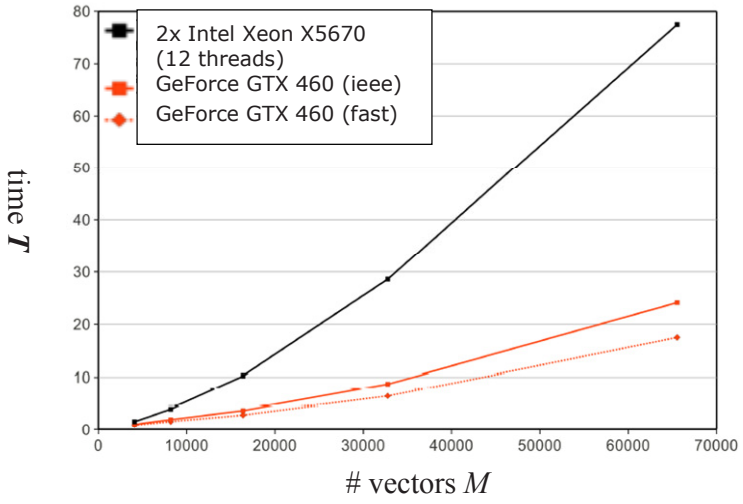

Fig.7. Timings comparison between CPU (OpenMP - 12 threads) and GPU (CUDA) implementations of SUBSET algorithm for H64 dataset (see Table 1).

We have tested the algorithm using several various data sets both artificially generated and taken from UCI repository [9]. The parallel implementation of SUBSET run on 12 cores of 2 Intel Xeon X5670 CPU board in OpenMP environment give the speedup between 8.5 to 10, compared to its optimized serial version. In Fig.6 we present typical results representing the values of stress function F(.) and timings as function of dataset size M for particle based MDS with full distance matrix (FULL) compared to SUBSET, RANDOM and GLIMMER

algorithms [11]. All of them were run on 12 threads of 2xIntel Xeon X5670 CPU board and were optimized in OpenMP environment. As shown in Fig.6, the performance of algorithms employing incomplete distance matrix (the same fraction of 2% distances were selected for each case) are very similar. However, GLIMMER algorithm reaches the highest discrepancy error.

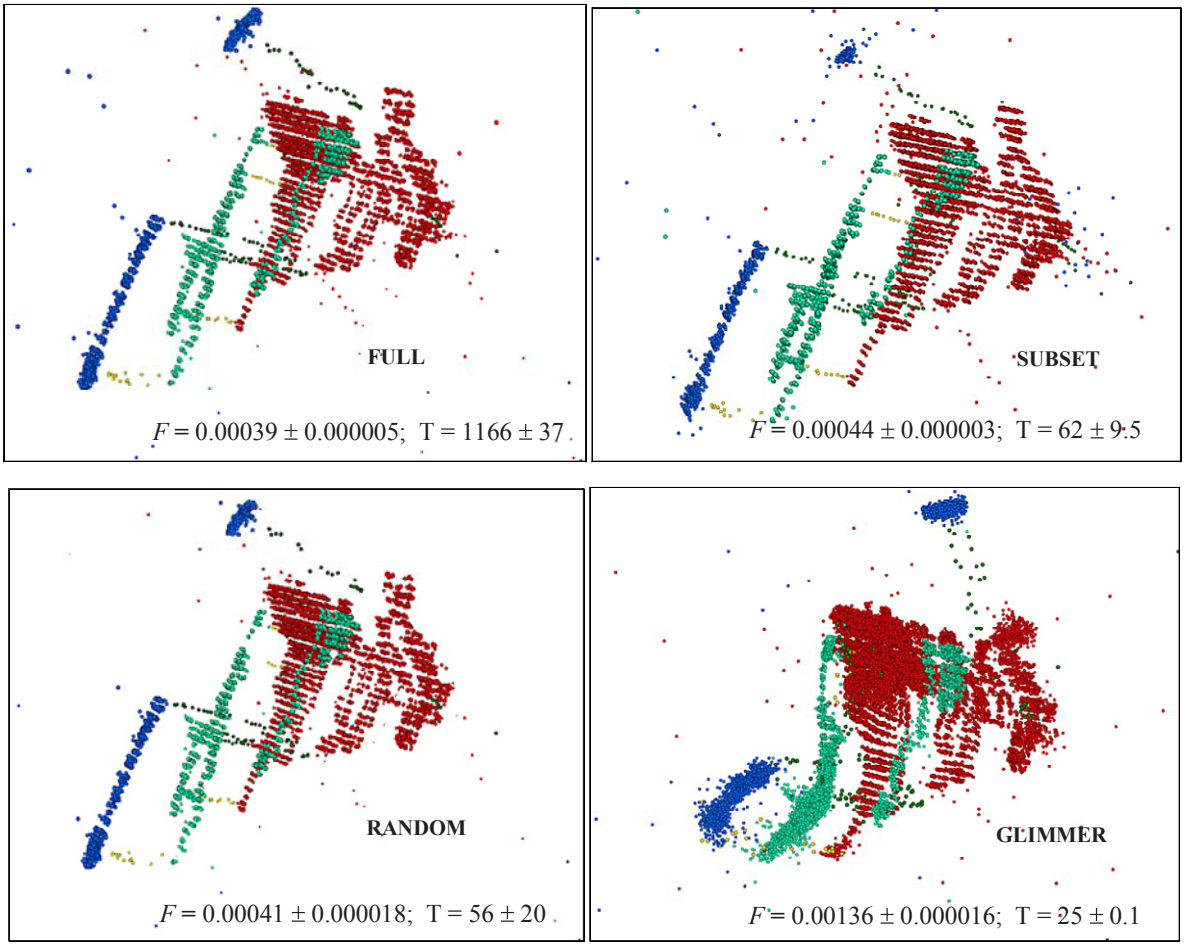

Fig.8. The results of MDS mapping of the "shuttle" dataset from UCI repository [9] (number of vectors M=0.58×10^5; dimensionality N=9) into 3-D target Euclidean space by using particle based MDS with full distance matrix (FULL), RANDOM, SUBSET and GLIMMER methods, respectively. The values of stress function $F(.)$ (1) (w=*1*, k=*1*, m=1) and simulation time T (in seconds) on 2xIntel Xeon X5670 are shown.

The advantage of SUBSET over RANDOM algorithm is evident when implemented on GPU board in CUDA environment. As shown in Fig.7, SUBSET algorithm is typically 3-4 times faster on GeForce GTX 460 board than on multi-core CPU. Meanwhile, the RANDOM algorithm is much slower on GPU than on CPU due to the lack of data locality. We have tested also the GPU version of GLIMMER. This GPU tuned version of MDS is more than 2 times faster than SUBSET. However, the discrepancy errors attained by GLIMMER are much greater than those obtained by SUBSET. Its weakness is demonstrated in Fig.8, where we show typical results of MDS mapping using particle based MDS (FULL, RANDOM and SUBSET algorithms) and GLIMMER algorithm. It is evident that the result of mapping is much better for our algorithm reproducing almost perfectly the best, i.e., giving minimal discrepancy error (FULL), MDS mapping result.

4. Conclusions

To decrease the high computational complexity of classical multidimensional scaling algorithms we have developed a novel MDS algorithm SUBSET which uses incomplete dissimilarity matrix Δ for multidimensional data visualization**.** It is dedicated for efficient exploitation of multi-thread processors architectures. We have demonstrated that SUBSET, which employs particle method for the error function minimization, can be used as the robust computational engine for interactive visualization of relatively large datasets consisting of $M \sim 10^5$ data items on up-to-the-date personal computers equipped with standard multi-core CPU and GPU boards. By using only a fraction (a few percent of order) of distances between data items, a small increase of the MDS discrepancy error, compared to that obtained by MDS with full distance matrix, is observed. The regular structure of distance matrix used in SUBSET algorithm allows for increasing its performance 4 times on GeForce 460 GTX GPU board in comparison to its OpenMP parallel version run on 12 cores of 2xIntel Xeon X5670 CPU board. Though slower than GPU tuned GLIMMER algorithm [11], SUBSET attains unbeatable small discrepancy error and allows for considerably better reconstruction of the source data space than GLIMMER.

Acknowledgements. This research is supported by the Polish Ministry of Higher Education and Science in scope of the project NN519 443039 and by AGH grant No.11.11.120.777. We would like to thank Nvidia Company for donating the Authors with Tesla C1060 GPU. For tests we used public domain source code of GLIMMER downloaded from the web page: http://www.cs.ubc.ca/~sfingram/glimmer/. The UCI Machine Learning Repository is kindly acknowledged for providing us with test data sets used in this work.

References

[1] Young G., Householder A. S. Discussion of a set of points in terms of their mutual distances. Psychometrika, 3(1):19-22, 1938.
[2] Torgerson W. S., Multidimensional scaling. 1. Theory and method, Psychometrika. 17:401-419, 1952.
[3] Kruskal J., Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. Psychometrika 29:1-27, 1964.
[4] Dzwinel W., Błasiak J. Method of particles in visual clustering of multi-dimensional and large data sets. Future Generation Computer Systems 15:365-379, 1999.
[5] Dzwinel W. Virtual particles and search for global minimum. Future Generation Computer Systems, 12:371-389, 1997.
[6] Kurdziel, M., Boryczko, K., Dzwinel W. Procrustes analysis of truncated least squares multidimensional scaling, Computing and Informatics. 31:1014-1440, 2012.
[7] Pawliczek, P., Dzwinel, W., Yuen, DA. Comparison of CPU versus GPU in Multidimensional Scaling for Large Data Sets. J. Concurrency: Practice and Experience*,* submitted December 2012
[8] Seung-Hee Bae, Judy Qiu and Fox, G. High Performance Multidimensional Scaling for Large High-Dimensional Data Visualization. IEEE Transaction of Parallel and Distributed System, January 2012, (in press)
[9] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/.
[10] De Silva V., Tenenbaum J.B. Global versus local methods in nonlinear dimensionality reduction. Advances in Neural Information Processing Systems, 15:705-712, 2003.
[11] Ingram S., Munzner T., Olano M., Glimmer: Multilevel MDS on the GPU. IEEE Transactions on Visualization and Computer Graphics, 15:249-261, 2009.
[12] Bronstein M.M., Bronstein A.M., Kimmel R., Yavneh I. Multigrid multidimensional scaling. Numerical Linear Algebra with Applications, 13:149-171, 2006.
[13] Chalmers M. A linear iteration time layout algorithm for visualising high dimensional data. IEEE Visualization '96. Proceedings, 127-131, 1996.