INFORMATICA, 2019, Vol. 30, No. 2, 349–365

DOI: http://dx.doi.org/10.15388/Informatica.2019.209

Hyperspectral Image Classification Using Isomap with SMACOF

Francisco José ORTS GÓMEZ[1]*, Gloria ORTEGA LÓPEZ[2],
Ernestas FILATOVAS[3], Olga KURASOVA[3],
Gracia Ester Martın GARZÓN[1]

[1] *Group of Supercomputation-Algorithms, Department of Informatics, University of Almería, ceiA3, 04120, Almería, Spain*

[2] *Computer Architecture Department, Campus Teatinos, Universidad de Málaga, 29010, Málaga, Spain*

[3] *Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, LT-08663, Vilnius, Lithuania*

e-mail: francisco.orts@ual.es, gmartin@ual.es, gloriaortega@uma.es, ernest.filatov@gmail.com, olga.kurasova@mii.vu.lt

Received: October 2018; accepted: January 2019

Abstract. The isometric mapping (Isomap) algorithm is often used for analysing hyperspectral images. Isomap allows to reduce such hyperspectral images from a high-dimensional space into a lower-dimensional space, keeping the critical original information. To achieve such objective, Isomap uses the state-of-the-art MultiDimensional Scaling method (MDS) for dimensionality reduction. In this work, we propose to use Isomap with SMACOF, since SMACOF is the most accurate MDS method. A deep comparison, in terms of accuracy, between Isomap based on an eigendecomposition process and Isomap based on SMACOF has been carried out using three benchmark hyperspectral images. Moreover, for the hyperspectral image classification, three classifiers (support vector machine, k-nearest neighbour, and Random Forest) have been used to compare both Isomap approaches. The experimental investigation has shown that better classification accuracy is obtained by Isomap with SMACOF.

Key words: dimensionality reduction, hyperspectral imaging, isometric mapping (Isomap), manifold learning, SMACOF algorithm.

1. Introduction

HyperSpectral Images (HSIs) contain an exhaustive variety of information about specific characteristics of the materials, with hundreds or even thousands bands (Borengasser *et al.*, 2007). The spectrum of each pixel can be seen as a vector, where each component represents the luminosity of the reflectance value for each spectral band. The set of bands which composes an HSI shows the representation of a scene, but each one individually contains information from a different wavelength range, which can cover both the visible

*Corresponding author.

and infrared spectrum. The width of each band can be between 5 and 10 nm, depending on the considered sensor. Each material throws a different reflectance profile for all the bands. Thus, for each point of the image, a specific curve that provides a lot of information for the corresponding point of the scene is obtained. Therefore, to efficiently exploit this information in applications, classification of HSIs is usually performed, where pixels are labelled to one of the classes based on their spectral characteristics.

There are many applications which take advantage of a large amount of information provided by hyperspectral sensors, such as remote sensing (Wang *et al.*, 2017), biotechnology (Asaari *et al.*, 2018), medical diagnose (Leavesley *et al.*, 2018), forensic science (Almeida *et al.*, 2017), environmental monitoring (Virlet *et al.*, 2017), etc. This available information leads us to develop new processing techniques. In addition, many applications which work with HSIs require a fast response. Examples of these applications may be obtained in the areas of modelling and environmental assessment, detection of military objectives or prevention and response to risks, such as forest fires, rescue operations, floods or biological threats (Chang *et al.*, 2001; Manolakis *et al.*, 2003).

However, the large amount of information contained in an HSI, which is its main advantage, is also a disadvantage in terms of computational performance. The work with large HSIs involves a high computational complexity and requires a lot of resources and time (Rizzo *et al.*, 2005). On the other hand, it is well-known that high-dimensional data spaces are mostly empty. This indicates that the data structure of an HSI exists basically in a subspace (Plaza *et al.*, 2005). Taking into account these ideas, it can be concluded that there is a need (and a possibility) to reduce the size of the HSIs. So, it is usual to apply techniques to reduce the dimensions of the original HSIs, obtaining reduced images which can be handled in a more efficient way without losing critical information (Harsanyi and Chang, 1994; Bruce *et al.*, 2002; Wang and Chang, 2006).

Multidimensional Scaling (MDS) consists of a set of techniques which are used to reduce the dimensions of a data set. Such techniques are used in many applications – multiobjective optimization (Filatovas *et al.*, 2015), data mining (Medvedev *et al.*, 2017), (Bernatavičienė *et al.*, 2007), marketing (Green, 1975), cryptography (Gupta and Ray, 2015), a wide variety of mathematical and statistical methods (Granato and Ares, 2014), psychology (Rosenberg, 2014), etc. They use a mapping function usually based on Euclidean distances which is able to find an optimal data representation. However, also other distance metrics could be considered (Fletcher *et al.*, 2014). MDS techniques represent data in a low-dimensional space in order to make these data more accessible (Borg and Groenen, 2005; Dzemyda *et al.*, 2013). For instance, a graphical visualization of the data in $2D$ or $3D$ space for an easier understanding of the information.

A well-known technique named Isometric mapping (Isomap) generalizes MDS to non-linear manifolds, replacing Euclidean distances by geodesic distances (Bengio *et al.*, 2004). Isomap has been used successfully in a multitude of applications, such as HSIs (Li *et al.*, 2017), face recognition (Yang, 2002a), biomedical datasets (Lim *et al.*, 2003), pattern classification (Yang, 2002b), learning multi-class manifold (Wu and Chan, 2004), supervised learning (Pulkkinen *et al.*, 2011), etc. Focusing on the HSIs, Isomap could be used in their reductions, achieving images with almost the same accuracy than the original but with fewer bands (Li *et al.*, 2017). The main goal here is to reduce the number of bands keeping the critical information they contain. Isomap is able to find hidden patterns in the bands and to reproduce the same pattern but with less bands.

Isomap often uses classical scaling such as eigen-decomposition as a part of its process. Classical scaling is a MDS method to reconstruct a configuration from the inter-point distance, which achieves a good accuracy and has a feasible computing cost (Sibson, 1979). However, any MDS method could be used.

The main contribution of this paper is the use of Isomap based on SMACOF (Scaling by MAjorizing a COmplicated), which is considered to be the most accurate MDS method (Borg and Groenen, 2005), and used when solving various MDS problems in social and behavioural sciences, marketing, biometrics, and ecology. Nevertheless, it is also one of the most computationally demanding methods (Ingram *et al.*, 2009). In previous work (Li *et al.*, 2017), where Isomap is studied in depth, authors consider classical scaling methods such as an eigen-decomposition process. However, our propose is to consider Isomap based on SMACOF due to its high accuracy. In this paper, the obtained results of both strategies, Isomap using eigen-decomposition and Isomap based on SMACOF, are compared in terms of classification accuracy. Such comparison is carried out by means of three popular HSIs and the same configurations in both cases.

The paper is organized as follows. In Section 2, the description of the Isomap method is provided. Section 3 describes the SMACOF algorithm. In Section 4, the results obtained after applying two versions of Isomap (with eigen-decomposition and with SMACOF) on several test images are discussed. Finally, we conclude this work in Section 5.

2. Isomap

Isomap is a manifold learning algorithm which can reduce the data redundancy preserving the original geometry of it. Isomap estimates the geodesic distance between all the items, given only input-space distances. For the points which are neighbours, input-space is an accurate approximation to the geodesic distance. For the distant ones, the geodesic distance can be computed as the addition of a sequence of distances between neighbouring points. The main idea is to find the shortest paths in a graph with edges connecting neighbouring data points (Tenenbaum *et al.*, 2000).

Isomap tries to build a matrix which contains all the minimum (geodesic) distances between the m items which are contained in a data set X (an HSI in our case), and then it reduces such matrix. In detail, the algorithm has three steps. They are shown in Algorithm 1 and described below:

1. To set a number l of neighbours. This number will be the same for all the items (points) X_i. Then, to determine the neighbours for every item X_i finding the l nearest points, taking into account that two points X_i and X_j cannot be neighbours if the distance between them is greater than a fixed value k. Euclidean distances between the m items are used. In this way, a graph G is constructed. Algorithm 2 describes the l-nearest neighbour (KNN) algorithm, which is commonly used to build neighbourhoods (Tay *et al.*, 2014).

Algorithm 1 Isomap($m, b, X, l, k, s, imax, \epsilon$)

```
Require:
    $m$: number of items;
    $b$: number of bands;
    $X$: $m \times b$ matrix which represents the HSI;
    $l$: maximum number of neighbours for each item;
    $k$: neighbourhood radius;
    $s$: dimension of low-dimensional space;
    $imax$: maximum number of iterations;
    $\epsilon$: threshold for the stress variance
Ensure:
    $Y$: set of finding points in the low-dimensional space stored in a $m \times s$ matrix
Construct the neighbourhood graph, $G = \text{KNN}(m, b, X, k, l)$ (Algorithm 2)
Compute shortest path between nodes, $\Delta = \text{Dijkstra}(m, G)$ (Algorithm 3)
Compute MDS method. For instance, SMACOF, $Y = \text{SMACOF}(m, s, \Delta, imax, \epsilon, Y)$ (Algorithm 4)
return $Y$
```

Algorithm 2 KNN(m, b, X, k, l, j)

```
Require:
    $m$: number of items;
    $b$: number of bands;
    $X$: $m \times b$ matrix which represents the HSI;
    $k$: neighbourhood radius;
    $l$: maximum number of neighbours for each item;
    $j$: index of the selected item
Ensure:
    $G$: The neighbourhood graph
for $i = 0; i < m; i++$ do
    Compute Euclidean distances, $D = [d(X_i, X_j)]$
Compute set $G$ containing indices for the $l$ smallest distances (with $l < k$) of each element of $D$
return $G$
```

2. To calculate the shortest distance between all pair of points in G. When X_i and X_j are neighbours, their distance is Euclidean. However, when the points are not neighbours, the distance is computed as the shortest path between all possible ones in G which connects X_i and X_j, that is, $d(X_i, X_j) = min\{d_G(X_i, X_j), d_G(X_i, X_n) + d_G(X_n, X_j)\}$, where $n = 1, \dots, m$. As a result of this step, an $m \times m$ matrix which contains the short distances Δ, is obtained. In this work, Dijkstra's algorithm has been used to calculate the shortest paths among G according to Algorithm 3 (Dijkstra, 1959). Authors in Deng *et al.* (2012) explain Dijkstra's algorithm as these steps:

   - To initialize all nodes to ∞, except the initial, which is set to 0. Neighbours already have their distances. To mark all nodes as unvisited, as it is shown in Fig. 1(a).
   - To consider all the unvisited neighbours and to calculate their distances through each node. For every neighbour, to compare this distance with its previous distance and to assign the smallest one to the node. An example is shown in Fig. 1(b).

Algorithm 3 Dijkstra(m, G)

```
Require:
    $m$: number of items;
    $G$: neighbourhood graph
Ensure:
    $\Delta$: $m \times m$ matrix with the shortest distances
for $i = 0; i < m; i{+}{+}$ do
    $\Delta[i] = \infty$
    $previous[i] = -1$
$\Delta[0] = 0$
$Q$ = the set of all nodes in $G$
while $Q$ is not empty do
    $u$ = vertex in $Q$ with the smallest $\Delta$
    if $\Delta[u] = \infty$ then break;
    Delete $u$ from $Q$
    for each neighbour $v$ of $u$ do
        $alt = \Delta[u] + distbetween(u, v)$
        if $alt < \Delta[v]$ then
            $\Delta[v] = alt$
            $previous[v] = u$
            Reorder $v$ in $Q$
return $\Delta$
```

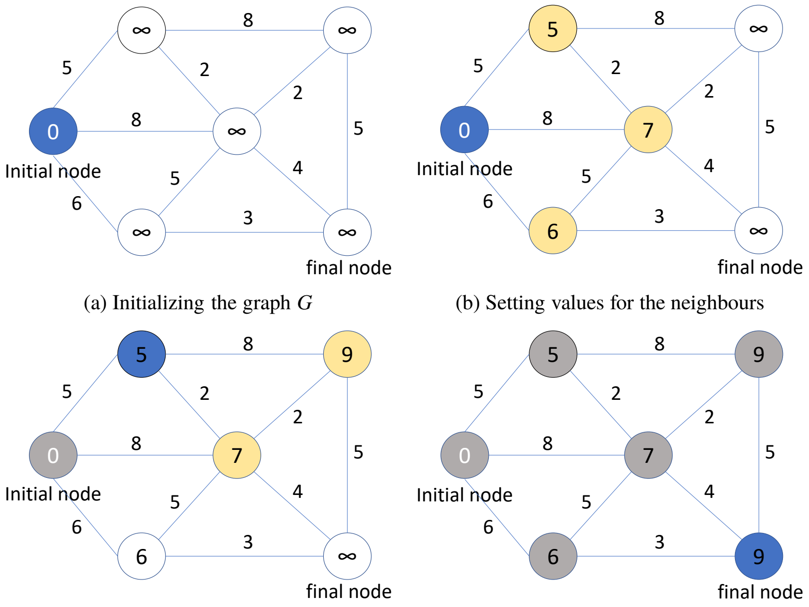
- When all the neighbours have been considered, to mark the current node as visited. A visited node will never be checked again. Move to the next unvisited node with the smallest distance and to repeat the previous steps, as it is shown in Fig. 1(c).
- If the final node has been marked as visited or if there is no path between the initial and the final node (all paths have a step marked as infinite), then the algorithm has finished. The final step is shown in Fig. 1(d).

3. To apply any MDS method to the shortest distances (Δ). Particularly, in this work, SMACOF and the eigen-decomposition methods are considered.

To evaluate the accuracy of Isomap based on SMACOF and eigen-decomposition methods for HSIs, a classification process with several classifiers – the Support Vector Machine (SVM) (Cortes and Vapnik, 1995), the KNN classifier (Altman, 1992) and the Random Forest algorithm (Breiman, 2001) – has been used.

3. SMACOF

SMACOF, as other MDS methods, is used for the analysis of similarity data on a set of items. As it has been mentioned before, SMACOF is the most accurate MDS technique (Ingram *et al.*, 2009). Its objective is to find a set of points $Y_1, Y_2, \ldots, Y_m \equiv Y$ in a low-dimensional space $\mathbb{R}^s$, $s < b$ (where b is the original number of dimensions), taking into account that the distances between these points must be as similar as possible to the distance between the original points $X_1, X_2, \ldots, X_m \equiv X$ (Orts *et al.*, 2018). The key is



(a) Initializing the graph *G*

(b) Setting values for the neighbours

(c) Moving to the nearest unvisited node

(d) Final step

Fig. 1. Steps of the Djikstra's algorithm.

the stress function (Eq. (1)). The less stress, the better results, since it measures the difference between the distances of the original points and the distances of the points in the low-dimensional space. In Eq. (1), δ represents the distance between points of X, and d does it between points of Y.

$$E_{MDS} = \sum_{i<j} \left(\delta_{ij} - d(Y_i, Y_j)\right)^2. \tag{1}$$

The majorizing concept, which implies to approximate a big or complex function through another smaller or simpler, is used by SMACOF to achieve the reduction of the stress (Groenen *et al.*, 1995). It consists of finding a new function iteratively. The new function will be located over the complex one, touching it at a point called *supporting* point (Fig. 2). Each iteration brings the minimum of the new function closer to the minimum of the original one, that is, the stress function (Borg and Groenen, 2005; Mairal *et al.*, 2014). In De Leeuw and Mair (2011), the majorization is defined in the following steps:

1. To choose an initial value $y = y_0$.
2. To find update x^t such that $g(x^t, y) \leqslant g(y, y)$.
3. If $f(y) - f(x^t) \geqslant \epsilon$, then $y = x^t$ and go to step 2.

In Algorithm 4, all the steps of SMACOF are shown. In such an algorithm, the initial value $y = y_0$ mentioned in step 1 is randomly generated. It has been tested in other works in which SMACOF obtains good results beginning from solutions randomly generated (Orts

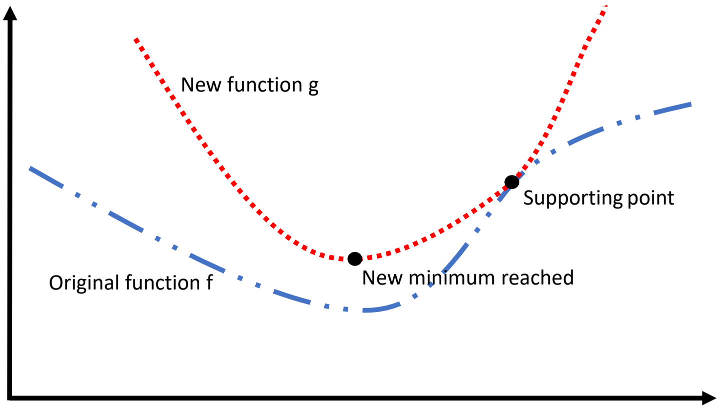


Fig. 2. Illustration of the majorization concept. The original function f is represented with a blue dashed line. The function obtained by majorization at every iteration, g, represented as a red dotted line, touches f at the supporting point. Taking into account that a new minimum of g is obtained at every iteration.

et al., 2018). The stress value of the current mapping is measured and then compared to the stress value of the previous mapping result. Each iteration minimizes the stress value due to the generation of closer solutions to the original. If the difference between the distances is smaller than a fixed threshold value, the algorithm stops (Ekanayake *et al.*, 2010), as it is mentioned in step 3. For the sake of simplicity, the details of the Guttman transform, used to update $x^{(t)}$, have not been explained here.

Algorithm 4 SMACOF($m, s, \Delta, imax, \epsilon$)

```
Require:
    $m$: number of items;
    $s$: dimension of low-dimensional space;
    $\Delta$: $m \times m$ matrix of dissimilarities of observed data on the high-dimensional space ($n$);
    $imax$: maximum number of iterations;
    $\epsilon$: threshold for the stress variance
Ensure:
    $Y$: set of finding points in the low-dimensional space stored in a $m \times s$ matrix
 1: Initial Solution randomly generated, $Y^0$
 2: Compute Euclidean distances, $D^0 = [d(Y_i^0, Y_j^0)]$
 3: $k = 0$, $error = 1$
 4: if ($k < imax$) and ($error > \epsilon$) then
 5:     Compute Guttman transform matrix, $B^k \equiv B^k(\Delta, D^{k-1})$
 6:     Compute Guttman transform, $Y^k = 1/m \cdot B^k \cdot Y^{k-1}$
 7:     Update distances $D^k = [d(Y_i^k, Y_j^k)]$
 8:     Compute $E_{MDS}^k$ (Eq. (1))
 9:     $error = |E_{MDS}^k - E_{MDS}^{k-1}|$
10:     $k = k + 1$
11: return $Y$
```

4. Evaluation Results

Such an investigation methodology has been considered in this work: first, to run Isomap based on SMACOF or eigen-decomposition methods and, after that, to apply a classification process with SVM, KNN or Random Forest classifiers.

Obtained results of Isomap using SMACOF are compared with the obtained results of a recent paper where Isomap considers an eigen-decomposition process (Li *et al.*, 2017) in the problem of hyperspectral images reduction. As in Li *et al.* (2017), three popular HSI images collected by the AVIRIS and ROSIS sensors have been considered to test Isomap (see Fig. 3). The considered data sets have the following characteristics:

- Pavia city centre (AVIRIS Salinas Valley, 2019), acquired by the ROSIS sensor. Pavia consists of 1096 × 715 pixels and 102 bands. For the sake of clarity, the data set is reduced to a 150 × 150 pixels subset. However, authors in Li *et al.* (2017) do not detail how they truncate the image in the study. In our work, random subsets of 150 × 150 are collected, keeping the ground truth variety.
- A finer spatial resolution of Salinas (AVIRIS sensor), named Salinas-A. Salinas-A consists of 86 × 83 pixels, which are the $[samples, lines] = [591 - 676, 158 - 240]$ of the original Salinas data set. It contains 204 bands.
- The Indian Pines data set (NW AVIRIS, 2012) collected by the AVIRIS sensor. It consists of 145 × 145 pixels and, originally, 224 bands. However, 24 bands which

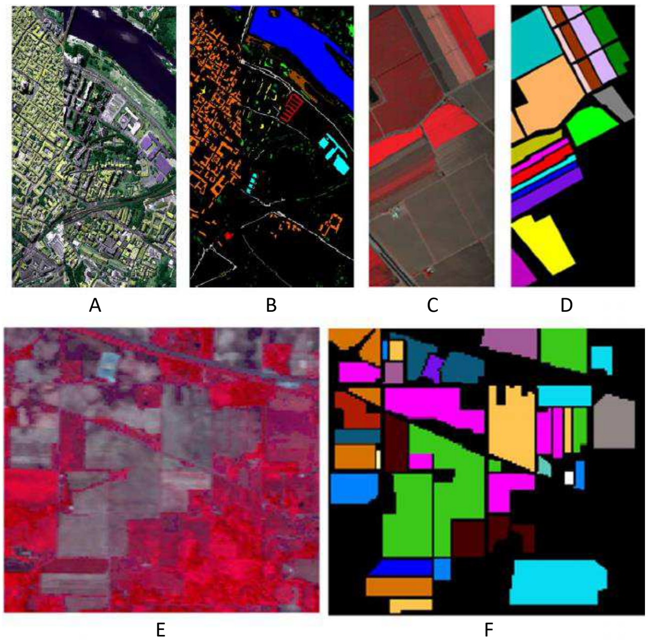


Fig. 3. HSIs tested. Pavia city centre (A) and its ground truth (B), Salinas-A (C) and its ground truth (D), and Indian Pines with its ground truth ((E) and (F) respectively).

contain the information about water absorption are removed in Li *et al.* (2017), so it has 200 bands in the tests.

Both Isomap versions (SMACOF and eigen-decomposition) have been implemented in Matlab and executed on a cluster composed by 64 cores of Bullx R424-E3 Intel Xeon E5 2650 with 8GB RAM. Specifically, KNN and Dijkstra procedures (Algorithms 2 and 3) have been coded using the Matlab functions *find_nn* and *dijkstra*, respectively. The precision of the classification process is dependent on the considered dimension of low-dimensional space (s) on Isomap. Therefore, several dimensions s have been taken into account to study their accuracy in the classification. Concretely, we varied the dimension of s from 10 to 50, as it was performed in Li *et al.* (2017). The parameter k, which describes the number of neighbours handled for each point has been set to 20.

We follow the idea described in Li *et al.* (2017) of considering several classifiers to evaluate the accuracy of both versions of Isomap for HSI classification, such as SVM and KNN classifiers. In addition, we have also considered the Random Forest algorithm. Similarly to Li *et al.* (2017), training and testing data were randomly selected from the ground truth. The 20% of the total pixels of each image were used to train, and the 80% to test. The comparative analysis has been based on the classification accuracy, which is obtained as the ratio: *correctly predicted data/total testing data*.

The SVM is coded using LIBSVM described in Chang and Lin (2011) with the following parameters: "$-t$ 2 $-c$ 100" ($-t$ 2 sets the type of kernel function as radial basis function, and $-c$ 100 set the cost parameter to 100). It is not necessary to set the gamma value, $-g$ (a parameter used as input by the radial basis function), as it is automatically set to "$-g\ 1/D$", where D is the dimension. The input data must be transformed following the data preprocessing described in Hsu *et al.* (2003). The results obtained using the SVM are depicted in Fig. 4.

KNN is a straightforward classification method, however, it is one of the most accurate ones (Keogh and Kasetty, 2002; Wei and Keogh, 2006). The results of the preliminary analysis of KNN are presented in Table 1 to consider the most suitable value of the number of neighbours (k'). This table shows the accuracy of the classification considering several values of k' $(1, 3, 5)$, for every reduced image on both dimensionality reduction methods (eigen-decomposition and SMACOF). Here, the best values are marked in italic style. As it can be observed in the table, the accuracy is reduced as the value of k' increases and 1NN obtains the best values of accuracy in all analysed cases. Therefore, KNN with $k' = 1$ (1NN) will be considered hereinafter. An additional advantage of 1NN is that it does not have tuning parameters and does not require a special transformation of the data or another preprocessing (Xing *et al.*, 2009). The Matlab function *fitcknn* has been used to perform KNN.

Apart from the classifiers used in Li *et al.* (2017), the Random Forest algorithm has also been considered in our evaluation (Fig. 6). The Matlab function *TreeBagger* has been used to perform Random Forest.

Obtained results with SVM, 1NN and Random Forest can be observed in Figs. 4, 5 and 6, respectively. The figures show the accuracy of the classification from the reduced images compared to the ground truth images, for both versions in each range from $s = 10$

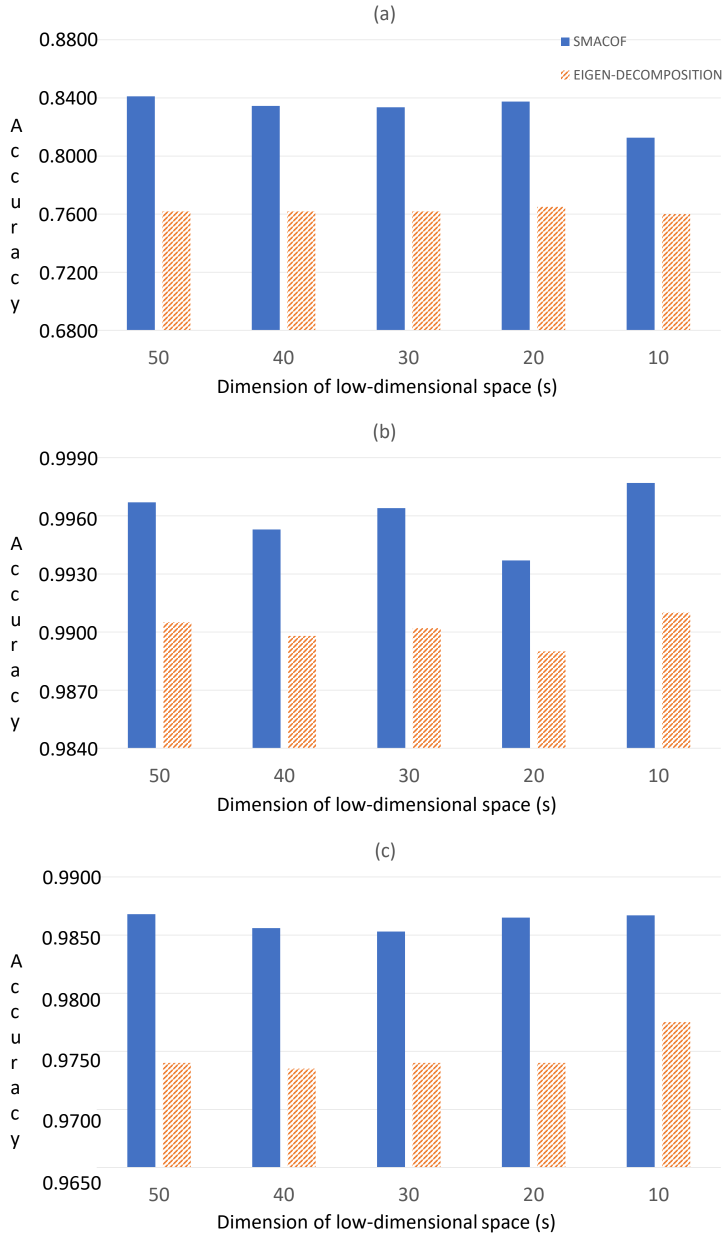


Fig. 4. Classification results (in terms of accuracy) of the three HSI data sets using SVM: (a) Indian Pines; (b) Salinas-A; (c) Pavia.

to $s = 50$. Such results have shown that the use of SMACOF improves the accuracy of Isomap for the three tested classifiers. In comparison with the version based on the eigen-decomposition process, the SMACOF approach is able to achieve better accuracies which involves a more optimized classification of HSI data sets.

Table 1
Classification results (in terms of accuracy) of the three HSI data sets using KNN for $k' = 1, 3$ and 5 and test images Indian Pines, Salinas-A and Pavia.

		SMACOF			EIGEN-DECOMPOSITION		
		k'					
IMAGE	s	1	3	5	1	3	5
Indian Pines	**50**	*0.8112*	0.7958	0.7943	*0.7250*	0.6956	0.6881
	40	*0.8046*	0.7987	0.7912	*0.7200*	0.6965	0.6884
	30	*0.8068*	0.7849	0.7814	*0.7150*	0.6933	0.6893
	20	*0.8179*	0.8069	0.7845	*0.7150*	0.6916	0.6879
	10	*0.8090*	0.7915	0.7877	*0.7050*	0.6896	0.6880
Salinas-A	**50**	*0.9946*	0.9931	0.9890	0.9899	0.9714	0.9658
	40	*0.9952*	0.9913	0.9925	*0.9896*	0.9733	0.9654
	30	*0.9950*	0.9935	0.9904	*0.9898*	0.9765	0.9645
	20	*0.9952*	0.9917	0.9914	*0.9892*	0.9743	0.9699
	10	*0.9963*	0.9890	0.9924	*0.9890*	0.9765	0.9687
Pavia	**50**	*0.9917*	0.9503	0.9488	*0.9729*	0.9365	0.9211
	40	*0.9929*	0.9407	0.9463	*0.9720*	0.9320	0.9232
	30	*0.9940*	0.9597	0.9525	*0.9729*	0.9365	0.9235
	20	*0.9937*	0.9598	0.9526	*0.9735*	0.9312	0.9245
	10	*0.9934*	0.9615	0.9576	*0.9715*	0.9348	0.9234

Once it is proven that the SMACOF approach is more accurate than the eigen-decomposition process, the global precision of Isomap with SMACOF has been tested in a more extended range of the values of s than (Li *et al.*, 2017) using 1NN (see Fig. 7). In this figure, it can be observed that the classification accuracy is quite high for all the analysed dimensionality reduction cases (from 50 to 2). However, it should be noted that the classification accuracy slightly decreases among the range from 9 to 2. Thus, we can conclude that SMACOF achieves a good accuracy even for the significant dimensionality reduction.

5. Conclusions

In this paper, our intention was to improve the accuracy of Isomap algorithm in the analysis of hyperspectral images. To achieve this, Isomap has been based on SMACOF, which is the most accurate MDS method, instead of classical scaling such as eigen-decomposition process.

The proposed version of Isomap based on SMACOF has been experimentally compared to a state-of-the-art version with an eigen-decomposition process. For that, well-known hyperspectral images taken from airbornes or satellites have been considered (Indian Pines, Salinas-A and Pavia Center). Moreover, a classification process using several classifiers (SVM, KNN and Random Forest) has been carried out to determine the accuracy of every test image with every method (SMACOF of eigen-decomposition). Obtained results have shown that the use of SMACOF improves the accuracy of Isomap in the reduction of the hyperspectral images for all studied cases.

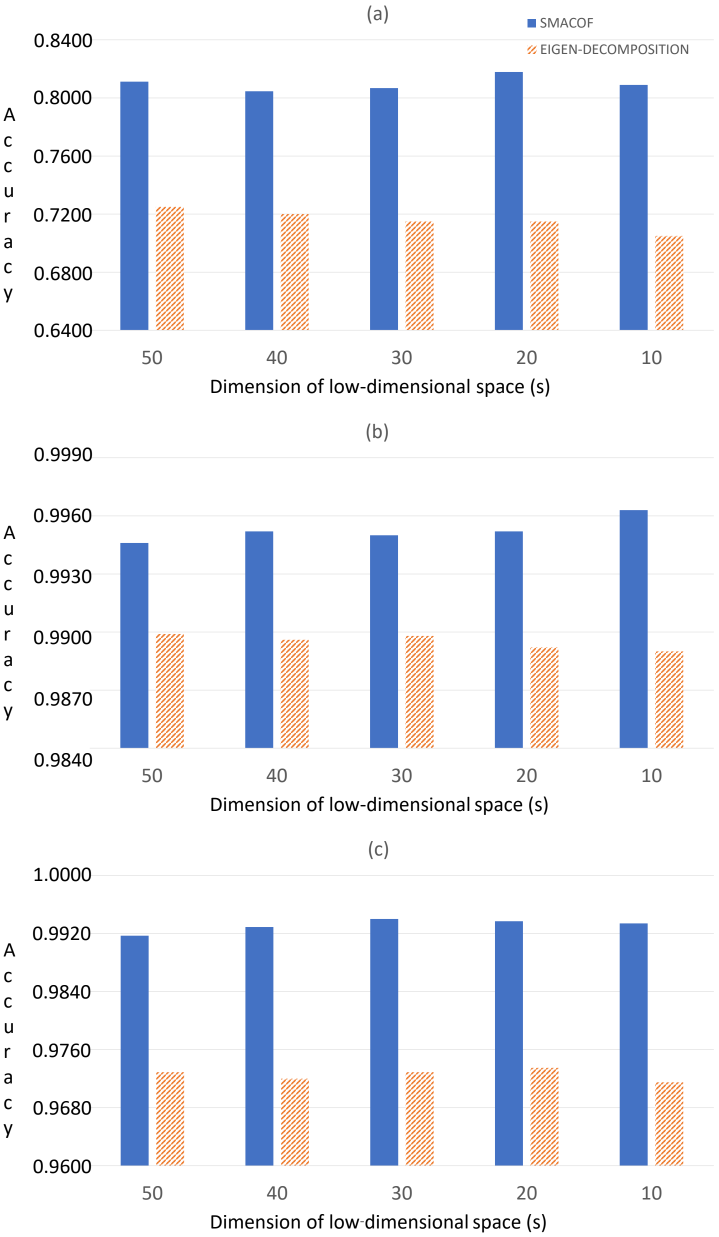


Fig. 5. Classification results (in terms of accuracy) of the three HSI data sets using 1NN: (a) Indian Pines; (b) Salinas-A; (c) Pavia.

In this work, only one criteria, the classification accuracy, is considered when reducing dimensions of the hyperspectral images. However, it should be noted that the drawbacks of Isomap and SMACOF are high consumptions of time and resources. Therefore, to de-

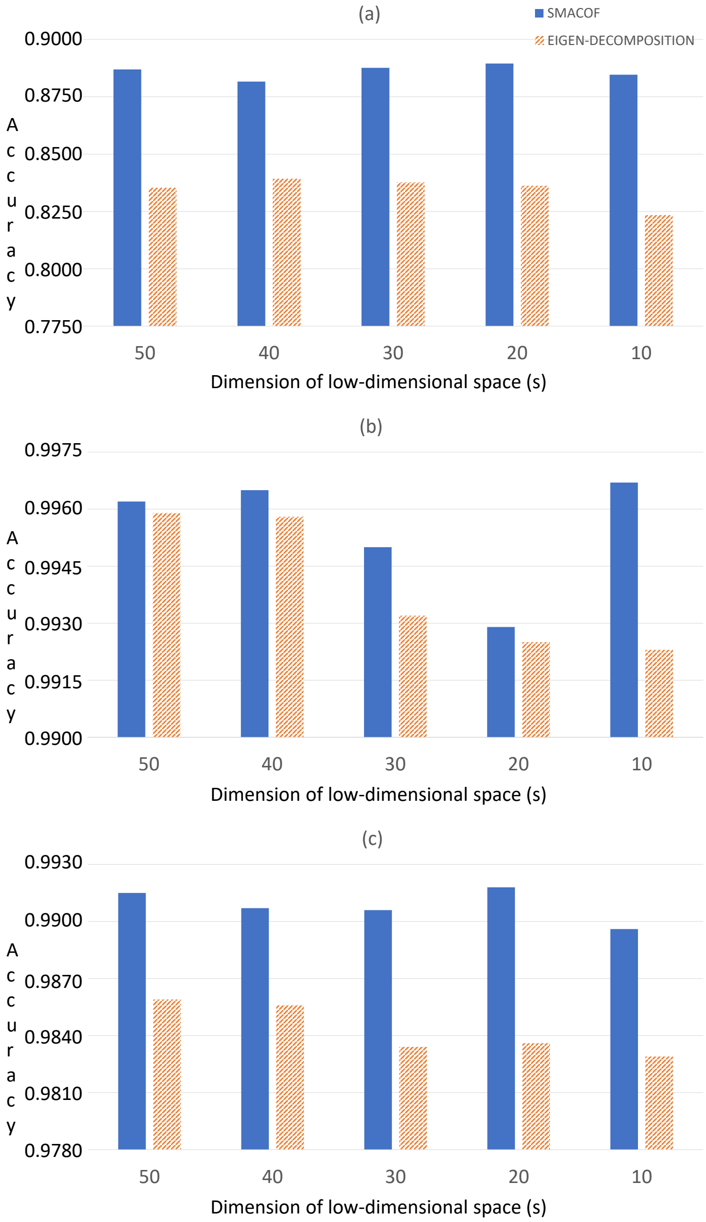


Fig. 6. Classification results (in terms of accuracy) of the three HSI data sets using Random Forest: (a) Indian Pines; (b) Salinas-A; (c) Pavia.

crease these aspects could be very valuable to make their application more approachable. Consequently, our current and future work is focused on the implementation of a GPU version of Isomap based on SMACOF.

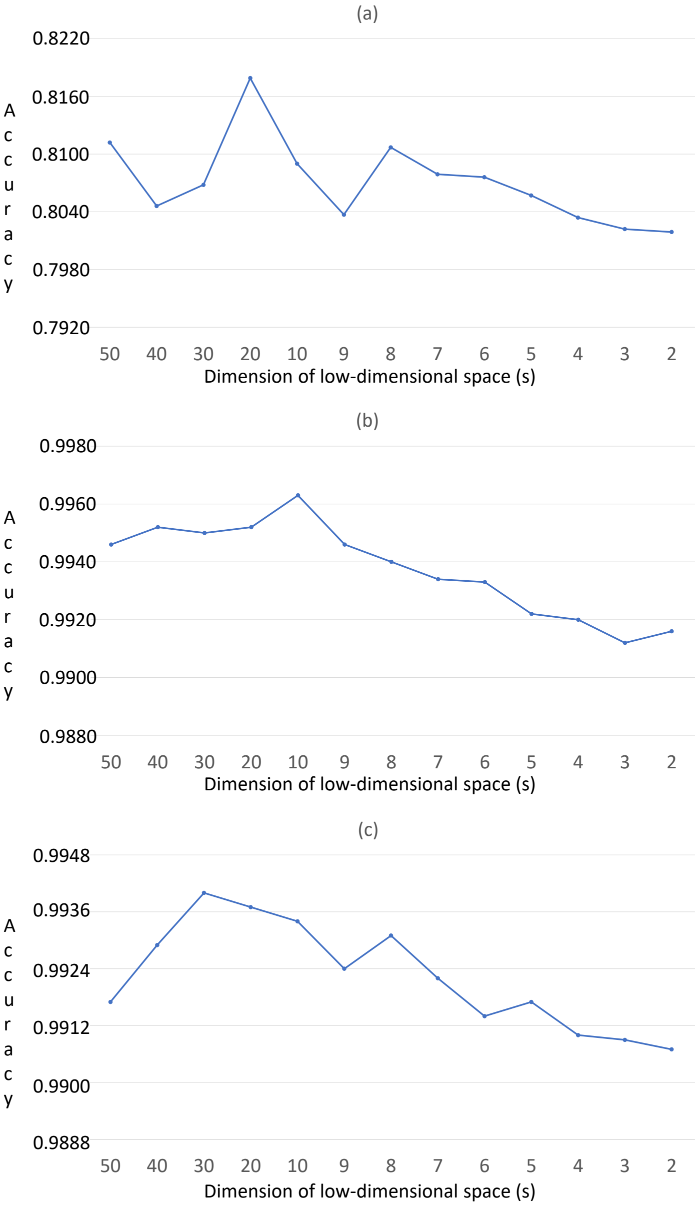


Fig. 7. Classification results (in terms of accuracy) of the three HSI data sets using 1NN for ranges from 50 to 2: (a) Indian Pines; (b) Salinas-A; (c) Pavia. Solid lines are to guide the eye.

Acknowledgements. This work has been supported by the Spanish Science and Technology Commission (CICYT) under contract TIN2015-66680; Junta de Andalucía under contract P12-TIC-301 in part financed by the European Regional Development Fund (ERDF). G. Ortega is a fellow of the Spanish ‘Juan de la Cierva Incorporación’ program.

References

Almeida, M., Logrado, L., Zacca, J., Correa, D., Poppi, R. (2017). Raman hyperspectral imaging in conjunction with independent component analysis as a forensic tool for explosive analysis: the case of an ATM explosion. *Talanta*, 174, 628–632.

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185.

Asaari, M., Mishra, P., Mertens, S., Dhondt, S., Inzé, D., Wuyts, N., Scheunders, P. (2018). Close-range hyperspectral image analysis for the early detection of stress responses in individual plants in a high-throughput phenotyping platform. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138, 121–138.

AVIRIS Salinas Valley (2019). Rosis pavia university hyperspectral datasets.

Bengio, Y., Paiement, J., Vincent, P., Delalleau, O., Roux, N., Ouimet, M. (2004). Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. In: *Advances in Neural Information Processing Systems*, pp. 177–184.

Bernatavičienė, J., Dzemyda, G., Marcinkevičius, V. (2007). Conditions for optimal efficiency of relative MDS. *Informatica*, 18(2), 187–202.

Borengasser M., Hungate W., Watkins R. (2007). *Hyperspectral Remote Sensing: Principles and Applications*. CRC Press.

Borg, I., Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media.

Breiman, L. (2001). *Random forests*. *Machine Learning*, 45(1), 5–32.

Bruce, L., Koger, C., Li, J. (2002). Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40(10), 2331–2338.

Chang, C., Ren, H., Chiang, S. (2001). Real-time processing algorithms for target detection and classification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(4), 760–768.

Chang, C., Lin, C. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.

Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

De Leeuw, J., Mair, P. (2011). *Multidimensional Scaling Using Majorization: Smacof in R*.

Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, 12(3), 1231–1237.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.

Dzemyda, G., Kurasova, O., Žilinskas, J. (2013). *Multidimensional Data Visualization: Methods and Applications*. Springer.

Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S., Qiu, J., Fox, G. (2010). Twister: a runtime for iterative mapreduce. In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM, pp. 810–818.

Filatovas, E., Podkopaev, D., Kurasova, O. (2015). A visualization technique for accessing solution pool in interactive methods of multiobjective optimization. *International Journal of Computers Communications & Control*, 10(4), 508–519.

Fletcher, R., Galiauskas, N., Zilinskas, J. (2014). Quadratic programming with complementarity constraints for multidimensional scaling with city-block distances. *Baltic Journal of Modern Computing*, 2(4), 248–259.

Granato, D., Ares, G. (2014). *Mathematical and Statistical Methods in Food Science and Technology*. John Wiley & Sons.

Green, P. (1975). Marketing applications of MDS: assessment and outlook. *The Journal of Marketing*, 24–31.

Groenen, P., Mathar, R., Heiser, W. (1995). The majorization approach to multidimensional scaling for Minkowski distances. *Journal of Classification*, 12(1), 3–19.

Gupta, K., Ray, I. (2015). Cryptographically significant MDS matrices based on circulant and circulant-like matrices for lightweight applications. *Cryptography and Communications*, 7(2), 257–287.

Harsanyi, J., Chang, C. (1994). Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4), 779–785.

Hsu, C., Chang, C., Lin, C. (2003). A practical guide to support vector classification, 1–16.

Ingram, S., Munzner, T., Olano, M. (2009). Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, 15(2), 249–261.

Keogh, E., Kasetty, S. (2002). On the need for time series data mining benchmarks: a survey and empirical demonstration. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 102–111.

Leavesley, S., Deal, J., Hill, S., Martin, W., Lall, M., Lopez, C., Boudreaux, C. (2018). Colorectal cancer detection by hyperspectral imaging using fluorescence excitation scanning. *Optical Biopsy XVI: Toward Real-Time Spectroscopic Imaging and Diagnosis*, 10489.

Li, W., Zhang, L., Zhang, L., Du, B. (2017). GPU parallel implementation of isometric mapping for hyperspectral classification. *IEEE Geoscience and Remote Sensing Letters*, 14(9), 1532–1536.

Lim, I., de Heras, P., Sarni, S., Thalmann, D. (2003). Planar arrangement of high-dimensional biomedical data sets by Isomap coordinates. In: *16th IEEE Symposium Computer-Based Medical Systems*, pp. 50–55.

Mairal, J., Bach, F., Ponce, P. (2014). Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2–3), 85–283.

Manolakis, D., Marden, D., Shaw, G. (2003). Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1), 79–116.

Medvedev, V., Kurasova, O., Bernatavičienė, J., Treigys, P., Marcinkevičius, V., Dzemyda, G. (2017). A new web-based solution for modelling data mining processes. *Simulation Modelling Practice and Theory*, 76, 34–46.

NW Aviris (2012). *Indianas indian pines 1992 data set*.

Orts, F., Filatovas, E., Ortega, G., Kurasova, O., Garzón, E.M. (2018). Improving the energy efficiency of smacof for multidimensional scaling on modern architectures. *The Journal of Supercomputing*, 1–13.

Plaza, A., Martinez, P., Plaza, J., Perez, R. (2005). Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 466–479.

Pulkkinen, T., Roos, T., Myllymaki, P. (2011). Semi-supervised learning for wlan positioning. In: *International Conference on Artificial Neural Networks*. Springer, pp. 355–362.

Rizzo, F., Carpentieri, B., Motta, G., Storer, J. (2005). Low-complexity lossless compression of hyperspectral imagery via linear prediction. *IEEE Signal Processing Letters*, 12(2), 138–141.

Rosenberg, S. (2014). The method of sorting in multivariate research with applications selected from cognitive psychology and person perception. *Multivariate Applications in the Social Sciences*, 123–148.

Sibson, R. (1979). Studies in the robustness of multidimensional scaling: perturbational analysis of classical scaling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 217–229.

Tay, B., Hyun, J., Oh, S. (2014). A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffusion tensor images. *Computational and Mathematical Methods in Medicine*.

Tenenbaum, J., De Silva, V., Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.

Virlet, N., Sabermanesh, K., Sadeghi-Tehran, P., Hawkesford, M. (2017). Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Functional Plant Biology*, 44(1), 143–153.

Wang, J., Chang, C. (2006). Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6), 1586–1600.

Wang, L., Zhang, J., Liu, P., Choo, K., Huang, F. (2017). Spectralspatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Soft Computing*, 21(1), 213–221.

Wei, L., Keogh, E. (2006). Semisupervised time series classification. In: *KDD 2006*, pp. 748–753.

Wu, Y., Chan, K. (2004). An extended Isomap algorithm for learning multi-class manifold. In: *Proceedings of 2004 International Conference IEEE Machine Learning and Cybernetics*, Vol. 6, pp. 3429–3433.

Xing, Z., Pei, J., Philip, S. (2009). Early prediction on time series: a nearest neighbor approach. In: *IJCAI*, pp. 1297–1302.

Yang, M. (2002a). Face recognition using extended Isomap. In: *IEEE Proceedings 2002 International Conference*.

Yang, M. (2002b). Extended Isomap for pattern classification. In: *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pp. 224–229.

F.J. Orts Gómez is a predoctoral researcher at the Informatics Department at University of Almería, Spain. He studied the master in computer engineering at the University of Almería. He is currently doing his PhD thanks to the Spanish FPI program. His publications and more information about him can be found in http://hpca.ual.es/~forts/. His research interests are multiDimensional scaling, quantum computation and high performance computing.

G. Ortega López (https://sites.google.com/site/gloriaortegalopez/) received the PhD degree from the University of Almería (Spain) in 2014. From 2009, she has been working as a member of the TIC-146 supercomputing-algorithms research group. Currently, she has a post-doctoral fellowship at the University of Málaga and her current research work is focused on high performance computing and optimization. Some of her research interest includes the study of strategies for load balancing the workload on heterogeneous systems, the parallelization of optimization problems and image processing.

E. Filatovas received the PhD in informatics engineering from the Vilnius University in 2012, Lithuania. He is currently a senior researcher at Vilnius University, and an associate professor at of Vilnius Gediminas Technical University. His main research interests include blockchain technologies, global optimization, multi-objective optimization, multi-objective evolutionary algorithms, multiple criteria decision making, high-performance computing, and image processing. He has published more than 20 scientific papers.

O. Kurasova received the doctoral degree in computer science (PhD) from Institute of Mathematics and Informatics jointly with Vytautas Magnus University in 2005. Recent employment is at the Institute of Data Science and Digital Technologies of the Vilnius University as a principal researcher and professor. Research interests include data mining methods, optimization theory and applications, artificial intelligence, neural networks, visualization of multidimensional data, multiple criteria decision support, parallel computing, image processing. She is the author of more than 70 scientific publications.

G.E.M. Garzón received her BSc degree in physics in 1985 from the University of Granada (Spain) and her PhD degree in computer engineering in 2000 from the University of Almería (Spain). She is a full-time professor at the Department of Informatics at the University of Almería. She is the head of the supercomputing-algorithms research group. Her research interest lies in the field of high performance computing for irregular problems related to image processing, matrix computation and optimization multicriteria.