Multidimensional Scaling

Amit Boyarski[1*] Alex Bronstein[1]

[1]Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel

1 Definition

Multidimensional scaling (MDS) refers to a family of techniques in data analysis that aim to realize a given matrix of *dissimilarities* $\boldsymbol{D}^{n\times n}$ (i.e., a distance matrix), as n points in a k-dimensional Euclidean space:

Find $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is as close as possible to d_{ij} $\forall ij$.

2 Background

The various multidimensional scaling models can be broadly classified into *metric* vs. *non-metric*, and *strain* (classical scaling) vs. *stress* (distance scaling) based MDS models. In metric MDS the goal is to maintain the distances in the embedding space as close as possible to the given dissimilarities, while in non-metric MDS only the order relations between the dissimilarities are important (i.e., $d_{ij} > d_{ql}$). *Strain* based MDS is an algebraic version of the problem that can be solved by eigenvalue decomposition. *Stress* based MDS uses a geometric distortion criterion which results in a non-linear and non-convex optimization problem. Each of these models has its own merits and drawbacks, both numerically and application-wise. On top of these basic models there exist numerous generalizations, including embedding into non-Euclidean domains, working with different stress models, working in different subspaces, and incorporating machine learning approaches to obtain faster, more accurate and more robust embeddings. The following sections will review these models, with emphasis on their role in computer vision applications.

Historically, MDS first appeared (informally) in the works of Dutch cartographer Jacob van Langren in the 17th century who was the first to show a table of distances alongside the map corresponding to these distances in a single figure [22]. The first modern version of MDS, called classical scaling, is due to

*Corresponding author. amitboy@cs.technion.ac.il

Torgerson who used it within the field of Psychophysics [41]. Later works by Shepard [39] and Kruskal [27, 28] formulated the problem as a minimization over a *stress* function, and introduced the first non-metric MDS model. A complete historical review can be found in [42]. A short review of the role of MDS in computer vision is given in Section 4.

3 Theory

3.1 Metric multidimensional scaling

3.1.1 Strain based scaling (classical scaling)

The Euclidean case. Let $\boldsymbol{D}^{n\times n}$ be a (symmetric) matrix of pairwise Euclidean distances between n points $\{\boldsymbol{y}_i\}_{i=1}^n$ in $\mathbb{R}^m$, and denote by $\boldsymbol{E}$ the matrix whose entries are the squares of these distances, i.e., $e_{ij} = d_{ij}^2 \equiv \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$. We have

$$\boldsymbol{E} = \mathbf{1}\boldsymbol{c}^\top + \boldsymbol{c}\mathbf{1}^\top - 2\boldsymbol{Y}\boldsymbol{Y}^\top \tag{1}$$

where $\boldsymbol{c}$ is a vector containing the diagonal elements of $\boldsymbol{Y}\boldsymbol{Y}^\top$. Assuming $n > m+2$, from (1) we have that the rank of $\boldsymbol{E}$ is $m+2$ [21]. Our goal is to find a set of points $\{\boldsymbol{x}_i\}_{i=1}^n$ in $\mathbb{R}^k$ such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \approx e_{ij}$. Let $\boldsymbol{H} \equiv \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ denote a *centering matrix*, i.e., multiplying a vector by this matrix results in a vector with zero mean. Applying $\boldsymbol{H}$ on both sides of a matrix is called a *double centering transform*, and results in a matrix whose mean along both columns and rows is zero. Since $\boldsymbol{H}\mathbf{1} = \left(\mathbf{1}^\top \boldsymbol{H}\right)^\top = \mathbf{0}$, we get that

$$\boldsymbol{K} \equiv -\frac{1}{2}\boldsymbol{H}\boldsymbol{E}\boldsymbol{H} = \boldsymbol{H}\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{H} \tag{2}$$

is a positive semi-definite Gram matrix of rank m. Classical (multidimensional) scaling aims at minimizing the following distortion criterion called *strain*

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times k}} \text{Strain}\left(\boldsymbol{X}\right) \equiv \min_{\boldsymbol{X}\in\mathbb{R}^{n\times k}} \left\|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{K}\right\|_F^2, \tag{3}$$

which amounts to finding the closest symmetric positive semi-definite matrix of rank $k \leq m$ to the matrix $\boldsymbol{K}$. Let $\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ be the spectral decomposition of $\boldsymbol{K}$. The solution of (3) is given by

$$\boldsymbol{X}^\star = \boldsymbol{V}_k\boldsymbol{\Lambda}_k^{\frac{1}{2}} \tag{4}$$

where $\boldsymbol{V}_k$ denotes the first k columns of $\boldsymbol{V}$, and $\boldsymbol{\Lambda}_k^{\frac{1}{2}}$ denotes a $k\times k$ diagonal matrix whose diagonal elements are the largest k eigenvalues of $\boldsymbol{K}$. The distortion can be quantified by summing over the smallest $m-k$ eigenvalues:

$$\text{Strain}\left(\boldsymbol{X}^\star\right) = \sum_{i=k+1}^{m} \lambda_i^2. \tag{5}$$

Note that

$$\boldsymbol{HEH} = \boldsymbol{E} - \frac{1}{n}\boldsymbol{E}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\boldsymbol{E} + \frac{1}{n^2}\mathbf{1}^\top\boldsymbol{E}\mathbf{1} \tag{6}$$

i.e., performing a double-centering transform does not require multiplication by a centering matrix on both sides, but rather can be done efficiently by subtracting the mean from each row and column and then adding back the mean of the entire matrix.

The non-Euclidean case. The above derivation assumed that the matrix $\boldsymbol{D}$ is a matrix of Euclidean distances. In real applications, this matrix represents a set of dissimilarities which might not be Euclidean distances or not even distances at all. A classical example is that of the perceptual distances between the RGB color channels in a digital image, as conceived by the human eye. A small change in the green channel is more noticeable than a similar change in the blue channel due to the eye's higher sensitivity to green. In these cases, there is no guarantee that the matrix $\boldsymbol{K}$ defined in (2) will be PSD, or even that it has k positive eigenvalues, as required by (4).

Numerical considerations. Despite (3) being a non-convex optimization problem, it enjoys a global solution in the form of an eigendecomposition. Nevertheless, it suffers from a few drawbacks:

- Classical scaling may ne impossible to obtain in case $\boldsymbol{D}$ is not a Euclidean distance matrix.
- Eigendecomposition techniques are expensive to apply and require the availability of the complete matrix $\boldsymbol{D}$, which may be time consuming to obtain in the best case, or impossible to obtain in the worst case.
- Classical scaling is very sensitive to outliers due to the double centering transform (2), which spreads the effect of a single outlier to the rest of the points in the same row/column.
- The strain distortion criterion relies on algebraic manipulations and lacks a geometric interpretation. This sometimes results in inferior embeddings (see for example [20]).

These drawbacks require the introduction of different MDS models and more delicate optimization methods. For example, iterative methods can be used for strain minimization, which allow the introduction of weights to handle missing entries in $\boldsymbol{D}$ [16], at the expense of the global convergence guarantees. Some of these approaches will be outlined in Section 3.4.

3.1.2 Stress based scaling (distance scaling)

As an alternative to strain minimization, consider the following non-linear optimization problem

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times k}} \sigma(\boldsymbol{X}) \equiv \min_{\boldsymbol{X}\in\mathbb{R}^{n\times k}} \sum_{i<j} w_{ij} \left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - d_{ij}\right)^2. \tag{7}$$

Here d_{ij} is a (symmetric) measure of dissimilarity between two sample points i and j; $w_{ij} \geq 0$ is a (symmetric) weight assigned to the pairwise term between those samples. The term $\sigma(\boldsymbol{X})$ is called Kruskal stress, LS-stress, ℓ_2-stress, or simply *stress*.

One efficient way to solve (7) is via the SMACOF algorithm [18]. First, note that

$$\sigma(\boldsymbol{X}) = \operatorname{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V} \boldsymbol{X}\right) - 2\operatorname{Tr}\left(\boldsymbol{X}^\top \boldsymbol{B}(\boldsymbol{X})\boldsymbol{X}\right) + \sum_{i<j} w_{ij} d_{ij}^2, \tag{8}$$

where $\boldsymbol{V}$ and $\boldsymbol{B}(\boldsymbol{X})$ are symmetric row-and-column-centered matrices given by

$$v_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k\neq i} w_{ik} & i = j \end{cases} \tag{9}$$

$$b_{ij} = \begin{cases} -w_{ij} d_{ij} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{-1} & i \neq j, \boldsymbol{x}_i \neq \boldsymbol{x}_j \\ 0 & i \neq j, \boldsymbol{x}_i = \boldsymbol{x}_j \\ -\sum_{k\neq i} b_{ik} & i = j, \end{cases} \tag{10}$$

and define

$$h(\boldsymbol{X}, \boldsymbol{Z}) = \operatorname{Tr}(\boldsymbol{X}^\top \boldsymbol{V} \boldsymbol{X}) - 2\operatorname{Tr}(\boldsymbol{Z}^\top \boldsymbol{B}(\boldsymbol{Z})\boldsymbol{X}) + \sum_{i<j} w_{ij} d_{ij}^2, \tag{11}$$

which is a convex (quadratic) function in $\boldsymbol{X}$ for a given $\boldsymbol{Z}$. By using the Cauchy-Schwarz inequality, it is possible to show that $h(\boldsymbol{X})$ is a *majorizing* function of $\sigma(\boldsymbol{X})$, i.e., satisfying (see [14])

$$h(\boldsymbol{X}, \boldsymbol{Z}) \geq \sigma(\boldsymbol{X}) \quad \forall \boldsymbol{X}, \boldsymbol{Z} \tag{12a}$$

$$h(\boldsymbol{X}, \boldsymbol{X}) = \sigma(\boldsymbol{X}). \tag{12b}$$

We now define the following iteration

$$\boldsymbol{X}_{k+1} = \operatorname*{argmin}_{\boldsymbol{X}} h(\boldsymbol{X}, \boldsymbol{X}_k) = \boldsymbol{V}^+ \boldsymbol{B}_k \boldsymbol{X}_k, \tag{13}$$

where $\boldsymbol{B}_k \equiv \boldsymbol{B}(\boldsymbol{X}_k)$, and $\boldsymbol{V}^+$ denotes the pseudo-inverse of the rank deficient matrix $\boldsymbol{V}$. By virtue of (12) and (13) it follows that

$$\sigma(\boldsymbol{X}_{k+1}) \leq h(\boldsymbol{X}_{k+1}, \boldsymbol{X}_k) \leq h(\boldsymbol{X}_k, \boldsymbol{X}_k) = \sigma(\boldsymbol{X}_k), \tag{14}$$

i.e., (13) is guaranteed to produce a sequence of monotonically non-increasing stress values.

The multiplicative update (13) suggests that at each iteration, each coordinate in the current embedding is a weighted mean of the coordinates of the embedding from the previous iteration, starting with an initial embedding $\boldsymbol{X}_0$, where the weights are given by the ratio of the pairwise distances, $d_{ij}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{-1}$.

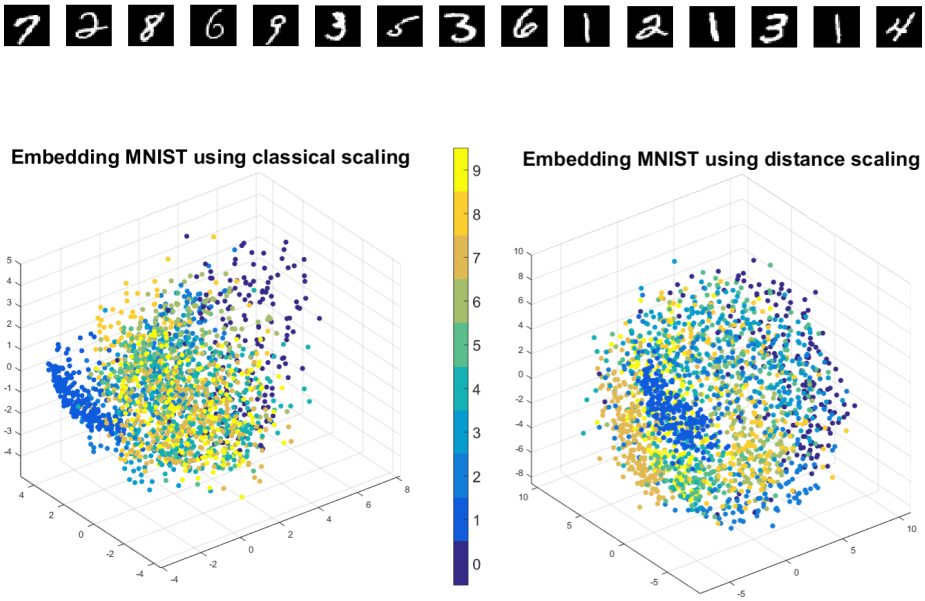

Figure 1: Top - a few examples of 28×28 images of handwritten digits from the MNIST dataset [1]. Bottom - embeddings in $\mathbb{R}^3$ of Euclidean distances between 2000 MNIST digits using classical scaling (left) and distance scaling (right).

Numerical considerations. Each iteration (13) of SMACOF requires the computation of the pairwise Euclidean distances between all points in the current embedding, a task of complexity $\mathcal{O}(n^2)$, and the solution of a linear system involving the matrix $\boldsymbol{V}$. Despite being rank deficient, $\boldsymbol{V}$ is not necessarily sparse, and computing its pseudo-inverse is of order $\mathcal{O}(n^3)$, except when $\boldsymbol{V}$ has a special form. For example, in case all the weights w_{ij} are equal to 1, we get

$$\boldsymbol{V}^+ = \frac{1}{n}\left(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right), \tag{15}$$

and since $\mathbf{1}^\top \boldsymbol{B} = \mathbf{0}$, (13) reduces to

$$\boldsymbol{X}_{k+1} = \frac{1}{n}\boldsymbol{B}_k \boldsymbol{X}_k. \tag{16}$$

3.2 Non-metric MDS

In non-metric MDS (nMDS), also called ordinal MDS, the goal is to preserve (as-best-as-possible) the *rank-order* of the distances in the embedding space

rather than their *value*. For example, for any four points i, j, q, l satisfying $d_{ij} > d_{ql}$ the goal is to find an embedding such that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| > \|\boldsymbol{x}_q - \boldsymbol{x}_l\|$. It is always possible to find an order preserving embedding of $\boldsymbol{D}$, using the following methodology [7]: Due to (2), the following relation holds between the elements of $\boldsymbol{K}$ and $\boldsymbol{E}$

$$\boldsymbol{e}_{ij} = k_{ii} + k_{jj} - 2k_{ij}. \tag{17}$$

The matrix

$$\boldsymbol{K'} = \boldsymbol{K} - \min\left(\lambda_{min}\left(\boldsymbol{K}\right), 0\right)\boldsymbol{I} \tag{18}$$

is always positive semi-definite, and can therefore be decomposed as $\boldsymbol{K'} = \boldsymbol{X}\boldsymbol{X}^\top$. We have

$$\begin{aligned} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= k'_{ii} + k'_{jj} - 2k'_{ij} \\ &= k_{ii} - \min\left(\lambda_{min}\left(\boldsymbol{K}\right), 0\right) + k_{jj} - \min\left(\lambda_{min}\left(\boldsymbol{K}\right), 0\right) - 2k_{ij} \\ &= e_{ij} - 2\min\left(\lambda_{min}\left(\boldsymbol{K}\right), 0\right). \end{aligned} \tag{19}$$

Thus, $\boldsymbol{X}$ is a Euclidean embedding of $\boldsymbol{E}$ with all entries shifted by a constant factor $2\min\left(\lambda_{min}\left(\boldsymbol{K}\right), 0\right)$, and thus all relative comparisons are preserved.

Despite this nice theoretical result, it may not serve as a practical algorithm since it usually results in high dimensional embeddings. For a more practical approach, we model the nMDS problem with the following distortion function, the Shepard-Kruskal stress, also called *stress*-1

$$\sigma_1\left(\boldsymbol{X}\right) = \min_{\theta} \frac{\sum_{i<j}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - \theta(d_{ij})\right)^2}{\sum_{i<j}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \tag{20}$$

where $\theta(\cdot)$ denotes an arbitrary monotonic transformation. Other definitions of non-metric stress functions exist. This model makes sense, for example, in Psychophysics, where the dissimilarities represent perceptual differences and their value is therefore rather arbitrary. The straight-forward way for minimizing (20) is by alternate minimization:

1. Fix θ and solve for $\boldsymbol{X}$, e.g., with gradient descent.
2. Minimize over θ using *isotonic regression*.

It is also possible to combine non-metric MDS with a strain based models, although it is quite unusual [16].

Numerical considerations

- Since (20) is not convex in $\boldsymbol{X}$, the first step may get caught in local minima.
- The above approach requires order relations between all pairs of points. These order relations might be contradictory, as they often come from human observations.

These problems can be addressed by a generalized nMDS scheme [7].

3.3 Acceleration schemes

The algorithms discussed above scale poorly due to slow convergence rate and quantities that require long computation times and large memory requirements. For example, the complexity of eigenvalue decomposition, as required by the classical scaling algorithm, is in general $\mathcal{O}(n^3)$. Computing a Euclidean pairwise-distance matrix between n points, as required in (10), takes $\mathcal{O}(n^2)$. Moreover, the SMACOF algorithm (13), being a first order optimization method, requires hundreds to thousands of iterations to converge. In order to accelerate MDS algorithms one can resort to two different strategies: *numerical acceleration* and *approximation*. In the first, dedicated optimization techniques are used in order to obtain faster convergence rates, without compromising the accuracy of the embedding. In the latter, a small compromise is allowed in order to achieve drastic savings in computation time and memory. These two approaches are often used in conjunction, where an approximated solution is successively refined as part of a larger numerical scheme.

In [25], Newton's method was explored as an alternative to the SMACOF algorithm for the minimization of (7). While the number of iterations until convergence drops dramatically, the cost per iteration becomes prohibitive for even moderate values of n, making second order methods impractical. Several works [19, 15, 10] propose to split the MDS problem into several sub-problems ("resolutions" or "scales") by embedding only a subset of carefully chosen points. The coarse embedding obtained from this sample set is then interpolated and refined. One can distinguish between *multiresolution* and *multigrid* approaches. The first solves each sub-problem independently before advancing to the next, while the second takes into account the higher-resolution sub-problem already when solving the lower-resolution one. Such a multigrid algorithm for MDS was described in [15]. The setting requires the availability of an additional structure, namely a discrete manifold that can be manipulated to obtain distances between points in different scales. Specially constructed decimation and interpolation operators allow moving between scales, but since these are ad-hoc constructions they are hard to generalize for different manifolds. A spectral approach was described in [10] which achieves better performance and generalizes easily across manifolds.

Another form of numerical acceleration relies on vector extrapolation techniques [40]. These techniques try to extrapolate the next iteration based on a few previous iterations, often leading to substantial acceleration at a very modest cost of some extra computations per iteration. [34] suggested to use the reduced rank extrapolation (RRE) variant in order to accelerate the SMACOF iteration for distance scaling. This technique indeed increases the convergence rate, but a few iterations of the full problem are still required in order to produce an expressible subspace of search directions. Being a meta-algorithm, it can be used together with other acceleration techniques. For more details see [14].

Subspace methods. Subspace methods restrict some of the quantities appearing in the MDS algorithms to a lower-dimensional linear subspace, thus achieving significant improvement in computation time. This reduced complexity is often traded for larger embedding distortion, but one has to keep in mind that the sought Euclidean embedding is usually just an intermediate step within some larger task which may overall benefit from this kind of implicit regularization. The different subspaces used vary between ad-hoc and algebraic to geometric constructions. Indeed, the MDS algorithms described above rely only on the provided dissimilarity matrix and completely ignore to the way these dissimilarities were obtained. For example, if these dissimilarities were obtained by measuring geodesic distances between points on a manifold $\mathcal{M}$, having access to this manifold opens up the possibility to refine the embedding by incorporating the local manifold structure into the optimization problem. A key tool in these techniques is the Laplace-Beltrami operator (LBO), an analogue of the Euclidean Laplacian for manifolds. In the discrete setting, this operator is given by $\boldsymbol{L} = \boldsymbol{A}^{-1}\boldsymbol{W}$, where $\boldsymbol{A}$ is a diagonal matrix with positive entries called *mass matrix*, and $\boldsymbol{W}$ is a symmetric positive semi-definite matrix whose rows and columns sum to zero, called *stiffness matrix*. These matrices are obtained from the discrete version of the manifold, for example - by the well known cotangent scheme [33]. The spectrum of the Laplacian can be obtained by solving the generalized eigenvalue problem $\boldsymbol{W}\boldsymbol{\phi} = \lambda\boldsymbol{A}\boldsymbol{\phi}$. The matrices containing the eigenvalues and eigenvectors of $\boldsymbol{L}$ shall be henceforth denoted by $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$. These act as non-Euclidean counterparts of "frequencies" and "Fourier basis", correspondingly.

Subspace approaches for classical scaling can be used to interpolate the dissimilarity matrix $\boldsymbol{D}$ from only a few entries d_{ij} $i, j \in \mathcal{I}$, as a preliminary stage to the standard classical scaling algorithm. [6] proposed to perform the interpolation by minimizing the Dirichlet energy

$$E_{Dirichlet}(\boldsymbol{D}) \equiv \langle \nabla \boldsymbol{D}, \nabla \boldsymbol{D} \rangle_{\mathcal{M}} = \operatorname{Tr}\left(\boldsymbol{D}^{\top}\boldsymbol{W}\boldsymbol{D}\boldsymbol{A}\right) + \operatorname{Tr}\left(\boldsymbol{D}\boldsymbol{W}\boldsymbol{D}^{\top}\boldsymbol{A}\right), \quad (21)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ denotes an inner product computed on the manifold. This kind of energy function is frequently used in computer vision as a means of promoting "smoothness" of signals. Representing $\boldsymbol{D}$ in the LBO eigenspace $\boldsymbol{\Phi}$,

$$\boldsymbol{D} = \boldsymbol{\Phi}\boldsymbol{\alpha}\boldsymbol{\Phi}^{\top} \quad (22)$$

leads to the following optimization problem

$$\min_{\boldsymbol{\alpha}} \operatorname{Tr}\left(\boldsymbol{\alpha}^{\top}\boldsymbol{\Lambda}\boldsymbol{\alpha}\right) + \operatorname{Tr}\left(\boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{\alpha}^{\top}\right) \quad \text{s.t.} \quad \left(\boldsymbol{\Phi}\boldsymbol{\alpha}\boldsymbol{\Phi}^{\top}\right)_{ij} = d_{ij}, \quad \forall i, j \in \mathcal{I}. \quad (23)$$

In practice, since $\boldsymbol{D}$ is a smooth function on $\mathcal{M}$, the subspace $\boldsymbol{\Phi}$ can be truncated, leading to drastic savings in computation time. The choice of this particular subspace is due to it being optimal for the interpolation of functions with bounded gradient norm on manifolds [4]. The combination of (21) together with (22) gives an additional benefit: the Hessian of (21) is diagonalized by $\boldsymbol{\Phi}$, leading to a simpler optimization problem. The hard constraints in (23) can also be

replaced by a penalty, leading to an even simpler problem that can be solved in closed form. The use of (22) in the classical scaling algorithm from Section 3.1.1 has been called *spectral MDS* [6]. A similar work suggested to work in the spatial domain, and use bi-Laplacian smoothing instead of the Dirichlet energy, leading to smoother and more accurate interpolations [35].

One prominent drawback of the spectral MDS framework is that the savings in time for distance computation due to the subspace representation do not translate to savings in computation time of the Euclidean embedding. Another drawback is that while the LBO eigenbasis is optimal for the interpolation of smooth functions, something better can be done when interpolating a *particular* function, for example - by using a basis extracted from the data itself. [38] proposed to use $q \ll n$ linearly independent columns of the squared-Euclidean distance matrix $\boldsymbol{E}$, encoded as a matrix $\boldsymbol{F}^{n\times q}$, and find a matrix $\boldsymbol{M}^{q\times n}$ such that

$$\boldsymbol{E} \approx \frac{1}{2}\left(\boldsymbol{F}\boldsymbol{M} + \boldsymbol{M}^\top \boldsymbol{F}^\top\right). \tag{24}$$

Recall that in the Euclidean case, the rank of $\boldsymbol{E}$ is $m+2$ where m is the dimension of the original Euclidean space, therefore such a decomposition is possible *without error*, assuming $q \geq m+2$. In the non-Euclidean case, $\boldsymbol{E}$ can still be approximated as a low-rank matrix with small error. [38] proposed to do so using the Nyström method, a standard numerical technique for interpolation. A key advantage of working directly on $\boldsymbol{E}$ instead of $\boldsymbol{D}$ is that one can perform classical scaling while keeping $\boldsymbol{E}$ in factorized form (24) saving both memory and time. For more details, see [37, 38].

Subspace approaches for distance scaling (i.e., stress based) restrict the embedding coordinates $\boldsymbol{X}$, or rather the displacement field $\boldsymbol{\delta} \equiv \boldsymbol{X} - \boldsymbol{X}_0$, to some linear subspace $\boldsymbol{\Phi} \in \mathbb{R}^{n\times p}$, i.e., $\boldsymbol{\delta} = \boldsymbol{\Phi}\boldsymbol{\alpha}$. In [10], the SMACOF iteration (13) was reformulated to the case where $\boldsymbol{\delta}$ is modeled as a p-bandlimited signal on some manifold $\mathcal{M}_0$, i.e., it can be approximated by a linear combination of the first p eigenvectors of the Laplace-Beltrami operator. This is justified by the fact that since we are looking for a distance preserving embedding, $\boldsymbol{\delta}$ should be smooth in the sense that close points on $\mathcal{M}_0$ should remain close on the final embedding. The new update step in terms of $\boldsymbol{\alpha}$ is given by

$$\boldsymbol{\alpha}_{k+1} = \left(\boldsymbol{\Phi}^\top \boldsymbol{V}\boldsymbol{\Phi}\right)^+ \boldsymbol{\Phi}^\top \left(\boldsymbol{B}_k \boldsymbol{X}_k - \boldsymbol{V}\boldsymbol{X}_0\right), \tag{25}$$

where $\boldsymbol{X}_{k+1} = \boldsymbol{X}_0 + \boldsymbol{\Phi}\boldsymbol{\alpha}_{k+1}$. Drawing analogy from the famous *noble identity* (Figure 2), it turns out to be enough to embed a subset of $q \approx 2p \ll n$ points to obtain the coefficients $\boldsymbol{\alpha}$,

$$\boldsymbol{\alpha}_{k+1} = \left(\boldsymbol{\Phi}^\top \boldsymbol{S}^\top \boldsymbol{V}^{\mathrm{s}} \boldsymbol{S}\boldsymbol{\Phi}\right)^+ \boldsymbol{\Phi}^\top \boldsymbol{S}^\top \left(\boldsymbol{B}^{\mathrm{s}}{}_k \boldsymbol{S}\boldsymbol{X}_k - \boldsymbol{V}^s \boldsymbol{S}\boldsymbol{X}_0\right), \tag{26}$$

where $\boldsymbol{V}^{\mathrm{s}}, \boldsymbol{B}^{\mathrm{s}}$ are the matrices defined in (9),(10) constructed only from the q sampled points, and $\boldsymbol{S}$ is a sampling matrix. The final embedding is obtained by $\boldsymbol{X}_* = \boldsymbol{X}_0 + \boldsymbol{\Phi}\boldsymbol{\alpha}_*$. The entire procedure is called *spectral SMACOF*. A single iteration of (26) consist of two steps: *Majorization* - the construction of $\boldsymbol{B}^{\mathrm{s}}(\boldsymbol{X}_k)$, which requires $\mathcal{O}(q^2)$ operations; and *Minimization* - the solution of

Figure 2: The noble identity states the conditions under which the order of downsampling and linear filtering can be exchanged in the Euclidean case. While the order can always be exchanged in the $\Rightarrow$ direction, the exchange in the $\Leftarrow$ direction is only possible if the filter has a special form: $h[n] = 0 \; \forall n \; : \lfloor n/M \rfloor \notin \mathbb{Z}$.

(26), which in general requires $\mathcal{O}(p^3)$ for the computation of $(\boldsymbol{\Phi}^\top \boldsymbol{S}^\top \boldsymbol{V}^{\mathrm{s}} \boldsymbol{S} \boldsymbol{\Phi})^+$ in the first iteration, and $\mathcal{O}(p^2)$ operations in the succeeding iterations. A similar approach was carried in [29] using a subspace constructed from the data, akin to PCA, for the purpose of visualizing multidimensional images. These modifications drastically reduce the computation time of the embedding, and allow the application of distance scaling to large scale problems or within iterative procedures that require multiple applications of the SMACOF algorithm.

Finally, it is possible to use a *non-linear* parametric model for the embedding, $f_{\boldsymbol{\Theta}}(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}^k$, where $\boldsymbol{\Theta}$ is a set of model parameters, whose number is much smaller than n. For example, [32] finds such a parametric mapping by training a neural-network to minimize the following loss

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i>j} \left(\| f_{\boldsymbol{\Theta}}(\boldsymbol{x}_i) - f_{\boldsymbol{\Theta}}(\boldsymbol{x}_j) \| - d_{ij} \right)^2 . \tag{27}$$

Embeddings found by such a non-linear parametric model result in smaller metric distortion per number of embedded points compared to their linear counterparts.

3.4 Generalizations

The literature on MDS is extensive, and along the years there have been generalizations of the basic schemes and algorithms in many ways. In this section we review a few of these generalizations, focusing on those oriented towards computer vision applications.

3.4.1 Variations on the stress theme

Robust classical scaling. As mentioned in Section 3.1.1, the double centering transform (2) makes the embedding produced by classical scaling very sensitive to outliers in the distance matrix. Moreover, if $\boldsymbol{D}$ is not a Euclidean distance matrix, (2) might not be positive semi-definite. To cope with these problems, a robust MDS model was proposed by [17]. Note that the following transformation applied to a PSD matrix $\boldsymbol{K}$ produces a squared-Euclidean distance matrix (see also (17)):

$$dist(\boldsymbol{K})_{ij} = k_{ii} + k_{jj} - 2k_{ij}. \tag{28}$$

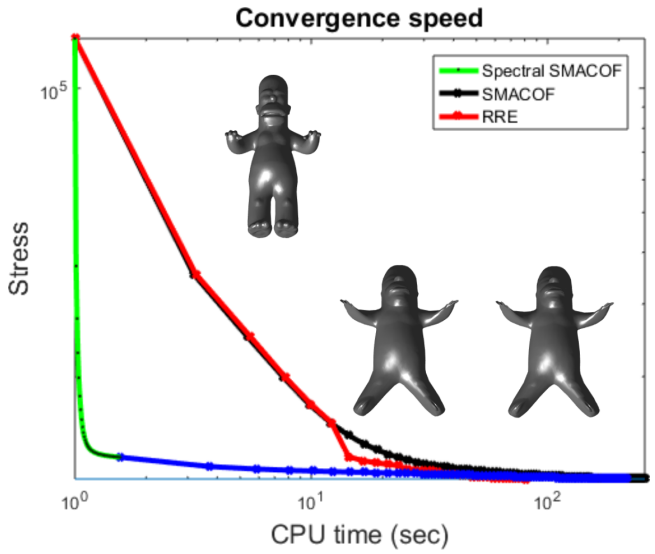

Figure 3: Runtime comparison between different numerical schemes for minimizing (7). The matrix $\boldsymbol{D}$ contains geodesic distances computed on the *homer* shape between 5103 points. The algorithms compared are the original SMACOF algorithm [18], an RRE vector extrapolation acceleration method [34] and spectral SMACOF [10]. By embedding only 200 points and extrapolating to the rest using the Lapalce-Beltrami eigenbasis (the green line), we achieve significant speedup compared to the other methods. The blue line represents additional SMACOF iterations at full resolution. The figures on the right are the original shape (top), and the results of spectral SMACOF (bottom-left) and regular SMACOF (bottom-right).

Consider the following optimization problem

$$\min_{\boldsymbol{K}\in\mathbb{R}^{n\times n}} \|\boldsymbol{E} - dist\,(\boldsymbol{K})\,\|_1 \;\text{ s.t. } \begin{cases} \boldsymbol{K} \succeq 0 \\ \operatorname{rank}(\boldsymbol{K}) \leq k. \end{cases} \tag{29}$$

(28) is a form of robust projection of the matrix $\boldsymbol{E}$ onto the set of matrices whose double-centering transform is PSD with rank at most k. The ℓ_1 norm handles the sparse set of outliers whereas the constraints make sure that the obtained matrix is embeddable into a k-dimensional Euclidean space. Once the solution is obtained, $dist(\boldsymbol{K})$ can be represented as $\boldsymbol{X}\boldsymbol{X}^\top$ as in the standard classical scaling algorithm.

Problem (28) can be solved by sub-gradient or interior point methods [17], or by combining the alternating direction method of multipliers (ADMM) with manifold optimization techniques [26]. The latter results in an efficient and simple algorithm that converges fast and scales well. To summarize this algorithm,

we first introduce an auxiliary variable $\boldsymbol{Z}$ and rewrite (28) as follows:

$$\min_{\boldsymbol{K},\boldsymbol{Z}} \|\boldsymbol{Z}\|_1 \text{ s.t. } \begin{cases} \boldsymbol{K} \succeq 0 \\ \operatorname{rank}(\boldsymbol{K}) \leq k \\ \boldsymbol{Z} = \boldsymbol{E} - dist(\boldsymbol{K}). \end{cases} \tag{30}$$

Introducing a (scaled) Lagrange multiplier $\boldsymbol{U}$ and a scalar μ, this problem can in turn be written in Lagrangian form

$$\min_{\boldsymbol{K},\boldsymbol{Z}} \|\boldsymbol{Z}\|_1 + \frac{\mu}{2} \|\boldsymbol{Z} - \boldsymbol{E} + dist(\boldsymbol{K}) + \boldsymbol{U}\|_F^2 \text{ s.t. } \begin{cases} \boldsymbol{K} \succeq 0 \\ \operatorname{rank}(\boldsymbol{K}) \leq k. \end{cases} \tag{31}$$

Now the standard ADMM method [12] can be invoked:

$$\boldsymbol{K}_{k+1} = \operatorname*{argmin}_{\boldsymbol{K}} \|\boldsymbol{Z}_k - \boldsymbol{E} + dist(\boldsymbol{K}) + \boldsymbol{U}_k\|_F^2 \text{ s.t. } \begin{cases} \boldsymbol{K} \succeq 0 \\ \operatorname{rank}(\boldsymbol{K}) \leq k \end{cases} \tag{32a}$$

$$\boldsymbol{Z}_{k+1} = \operatorname*{argmin}_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_1 + \frac{\mu}{2} \|\boldsymbol{Z} - \boldsymbol{E} + dist(\boldsymbol{K}_{k+1}) + \boldsymbol{U}_k\|_F^2 \tag{32b}$$

$$\boldsymbol{U}_{k+1} = \boldsymbol{U}_k + (\boldsymbol{Z}_{k+1} - \boldsymbol{E} + dist(\boldsymbol{K}_{k+1})) \tag{32c}$$

where the optimization over $\boldsymbol{B}$ can be carried out using manifold optimization techniques [3, 9], and the optimization over $\boldsymbol{Z}$ is an unconstrained non-smooth optimization problem which has a closed form solution using the shrinkage operator $T_\alpha(x) = (|x| - \alpha)_+ \, sign(x)$

$$\boldsymbol{Z}_{k+1} = T_{\frac{1}{\mu}}(\boldsymbol{E} - dist(\boldsymbol{K}_{k+1}) - \boldsymbol{U}). \tag{33}$$

Robust distance scaling. While the least-squares stress function defined in (7) is the most common one, it exhibits similar shortcomings to those often encountered when using the ℓ_2 norm in regression problems: the quality of the solution deteriorates quickly with any additional outlier added to the dataset. In order to cope with outliers, one can replace the ℓ_2 stress defined in (7) with a more robust version, say - the ℓ_1 stress

$$\sigma_{\ell_1}(\boldsymbol{X}) = \sum_{i<j} w_{ij} \left| \|\boldsymbol{x}_i - \boldsymbol{x}_j\| - d_{ij} \right|. \tag{34}$$

One simple heuristic algorithm for the minimization of (34), and in fact - of any other function of the form

$$\sigma_\rho(\boldsymbol{X}) = \sum_{i<j} \rho\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - d_{ij}\right) \tag{35}$$

is the *iteratively reweighted least squares* (IRLS) algorithm. The algorithm follows from a comparison between the sufficient condition of optimality for minimizing (7)

$$\nabla\sigma(\boldsymbol{X}) = \sum_{i<j} 2w_{ij} \left(\|x_i - x_j\| - d_{ij}\right) \nabla_{\boldsymbol{X}} \left(\|x_i - x_j\|\right) = 0 \tag{36}$$

and (35)

$$\nabla\sigma_\rho(\boldsymbol{X}) = \sum_{i<j} \rho'\left(\|x_i - x_j\| - d_{ij}\right) \nabla_{\boldsymbol{X}} \left(\|x_i - x_j\|\right) = 0. \tag{37}$$

By setting the weights in (36) to

$$w_{ij} = \frac{\rho'\left(\|x_i^* - x_j^*\| - d_{ij}\right)}{2\left(\|x_i^* - x_j^*\| - d_{ij}\right)} \tag{38}$$

we get an equivalent LS-MDS problem to (35). The IRLS algorithm works by solving a series of LS-MDS problems with iteratively updated weights:

1. set
$$w_{ij}^k = \frac{\rho'\left(\|x_i^k - x_j^k\| - d_{ij}\right)}{2\left(\|x_i^k - x_j^k\| - d_{ij}\right)} \tag{39}$$
2. solve
$$\boldsymbol{X}_{k+1} = \operatorname*{argmin}_{\boldsymbol{X}} \sum_{ij} w_{ij}^k \left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - d_{ij}\right)^2. \tag{40}$$

Another way to generalize distance scaling is by adding a regularization term, weighted by some constant $\mu > 0$, to the LS-stress

$$\min_{\boldsymbol{X}} \ \sigma(\boldsymbol{X}) + \mu R(\boldsymbol{X}). \tag{41}$$

This regularization term can help reject outliers, drive the embedding towards a favorable configuration or promote any other property that we want the embedding to manifest. Assuming the regularization function $R(\boldsymbol{X})$ is convex, a small variant of the SMACOF iteration described in Section 3.1.2 can be used in order to minimize the regularized stress. Instead of using the majorizing function $h(\boldsymbol{X}, \boldsymbol{Z})$ defined in (11), we shall use $h(\boldsymbol{X}, \boldsymbol{Z}) + \mu R(\boldsymbol{X})$. The update step (13) now requires the solution of a convex optimization problem

$$\boldsymbol{X}_{k+1} = \operatorname*{argmin}_{\boldsymbol{X}} h(\boldsymbol{X}, \boldsymbol{X}_k) + \mu R(\boldsymbol{X}). \tag{42}$$

3.4.2 Non-Euclidean embeddings

We shall conclude the section by reviewing a few schemes that consider embeddings to domains other than $\mathbb{R}^k$. The simplest choice of a non-Euclidean domain is a k-dimensional unit sphere $\mathcal{S}^m = \{\boldsymbol{Z} \in \mathbb{R}^{k+1} : \|z\|_2 = 1\}$. Embedding into a sphere can be formulated as follows (using the LS-stress)

$$\min_{\boldsymbol{Z}} \sum_{i<j} w_{ij} \left(d_{\mathcal{S}^m}(\boldsymbol{z}_i, \boldsymbol{z}_j) - d_{ij}\right)^2 \quad \text{s.t.} \ \ \|\boldsymbol{z}_q\| = 1 \ \ \forall q, \tag{43}$$

and can be solved by parametrizing $\boldsymbol{z}$ in spherical coordinates and using gradient descent. Embedding into a sphere with a different radius r amounts to scaling the dissimilarities by $1/r$.

[13] introduced *generalized MDS* (GMDS) - an algorithm for isometric embedding of one Riemannian manifold into another. The algorithm works as follows: given two such manifolds $\mathcal{X}, \mathcal{Y}$, discretized as triangular meshes, we compute the geodesic distances between all pairs of vertices on each of the meshes - $\boldsymbol{D}_{\mathcal{X}}$,$\boldsymbol{D}_{\mathcal{Y}}$. These distances can be used to interpolate the distance between any two points on the faces of the meshes. The embedding of one manifold into the other is represented in barycentric coordinates $\boldsymbol{U}^{n\times 3}$, and the generalized stress

$$\sigma_{GMDS}(\boldsymbol{U}) = \sum_{i<j} \|d_{\mathcal{X}}(\boldsymbol{u}_i, \boldsymbol{u}_j) - d_{\mathcal{Y}_{ij}}\|^2 \tag{44}$$

is minimized using gradient descent, subject to $\boldsymbol{U}\mathbf{1} = \mathbf{1}$. A spectral version of GMDS is developed in [5], following the spectral MDS algorithm [6].

Another form of non-Euclidean embedding is given in [36], who suggested to decompose the distance matrix $\boldsymbol{D}$ into $\boldsymbol{X}\boldsymbol{X}^\top$. Since $\boldsymbol{D}$ is symmetric, such a decomposition is always possible, based on the spectral decomposition of $\boldsymbol{D}$

$$\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Lambda}_D\boldsymbol{U}^\top. \tag{45}$$

$\boldsymbol{X} \equiv \boldsymbol{U}\boldsymbol{\Lambda}_D^{\frac{1}{2}}$ is an embedding of $\boldsymbol{D}$ into the complex domain $\mathbb{C}^n$ *without error*. Since most of the eigenvalues of $\boldsymbol{D}$ are of small magnitude [21], the metric distortion introduced by truncating the eigen-expansion is small, leading to highly accurate embeddings at the cost of working in the complex domain.

4 Applications

Multidimensional Scaling (MDS) is one of the most popular methods in data analysis, and it would be impossible to provide a complete set of applications in this short review. We will focus on a few applications of MDS in computer vision. Some of these applications are also detailed in Figure 4.

The main use of MDS algorithms is for dimensionality reduction. It is popularly used for visualization of high dimensional signals by embedding their dissimilarities into 2 or 3-dimensional Euclidean spaces. For example, [29] uses it to visualize multidimensional images, and [31] explores multiples visualization techniques on the MNIST dataset of handwritten digits [1] (Figure 1). Apart from visualization, low dimensional isometric embedding is used as a pre-process for algorithms that rely on Euclidean distances. Many classical image processing and computer vision algorithms have been developed to work on RGB color images. A simple way to adapt these algorithms to be used with high-dimensional features is to embed these features into a 3-dimensional Euclidean space, represented as a color image. For example, [30] uses MDS to embed features that integrate local color and gradient value distribution, and [24] uses it to embed the co-occurrence statistics of pixels in an image (Figure 4a). These embeddings are later used in classical segmentation and optical flow estimation algorithms.

Other image processing tasks such as image retargetting can be carried out efficiently with the subspace version of MDS [11], where the idea is to re-scale an image in a way that preserves distances in the salient regions, and concentrate the distortions in other parts, while preventing flip-overs.

In the field of geometry processing, MDS found applications in tasks such as texture mapping [43], shape analysis [20] and shape synthesis [8]. Classification of surfaces that is invariant to isometric transformation is achieved by embedding the surfaces to a Euclidean domain, creating for each surface a "canonical form" (Figure 4b). These canonical forms can be aligned with a rigid transformation, and a threshold on the alignment mismatch rules weather two shapes belong to the same class or not. This idea has been carried out using multiple versions of the MDS algorithms [20, 6, 35, 38, 36]. Alternatively, the generalized MDS scheme (44) [13] or its spectral version [5] can compare two surfaces directly, without passing through an intermediate embedding space. More recent works [23] use the GMDS cost (44) as a loss-function of a neural network that is trained in an unsupervised way to find optimal descriptors for shape matching.

In [8], MDS was used as part of the surface-from-operator problem. MDS plays a part in reconstructing a surface from its discrete metric, i.e., the lengths of the edges of the polygons comprising the mesh.

[7] developed a generalized non-metric MDS model, able to cope with missing and contradictory data, and used it for visualizing dissimilarities between reflectance maps of objects (Figure 4c). Since non-metric MDS models are mostly used to perform perceptual embeddings, they are rarely used in computer vision applications. Nevertheless, there is a great potential for using them in problems that rely only on rank-order of dissimilarities such as correspondence problems between non-isometric shapes.

5 Open Problems

One of the big and presently unsatisfactorily solved problems related to MDS methods is out-of-sample extension. We believe that more research into the use of modern machine learning tools, in particular, deep learning could bring new efficient solutions to this problem.

References

[1] The mnist database of handwritten characters. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2019-03-20.

[2] Toolbox for surface comparison and analysis. `http://tosca.cs.technion.ac.il/book/resources_data.html`.

[3] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[4] Yonathan Aflalo, Haim Brezis, and Ron Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015.

[5] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision*, 118(3):380–392, 2016.

[6] Yonathan Aflalo and Ron Kimmel. Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences*, 110(45):18052–18057, 2013.

[7] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18, 2007.

[8] Davide Boscaini, Davide Eynard, and Michael M Bronstein. Shape-from-intrinsic operator. *arXiv preprint arXiv:1406.1925*, 2014.

[9] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

[10] Amit Boyarski, Alex M Bronstein, and Michael M Bronstein. Subspace least squares multidimensional scaling. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 681–693. Springer, 2017.

[11] Amit Boyarski, Alex M Bronstein, and Michael M Bronstein. Subspace least squares multidimensional scaling. *arXiv preprint arXiv:1709.03484*, 2017.

[12] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[13] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.

[14] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[15] Michael M Bronstein, Alexander M Bronstein, Ron Kimmel, and Irad Yavneh. Multigrid multidimensional scaling. *Numerical linear algebra with applications*, 13(2-3):149–171, 2006.

[16] Andreas Buja, Deborah F Swayne, Michael L Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008.

[17] Lawrence Cayton and Sanjoy Dasgupta. Robust euclidean embedding. In *Proceedings of the 23rd international conference on machine learning*, pages 169–176. ACM, 2006.

[18] Jan De Leeuw, In JR Barra, F Brodeau, G Romier, B Van Cutsem, et al. Applications of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*. Citeseer, 1977.

[19] Vin De Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, pages 721–728, 2003.

[20] Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on pattern analysis and machine intelligence*, 25(10):1285–1295, 2003.

[21] John Clifford Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[22] PJ Groenen and Ingwer Borg. Past, present, and future of multidimensional scaling. *Visualization and verbalization of data*, pages 95–117, 2014.

[23] Oshri Halimi, Or Litany, Emanuele Rodolà, Alex Bronstein, and Ron Kimmel. Self-supervised learning of dense shape correspondence. *arXiv preprint arXiv:1812.02415*, 2018.

[24] Rotal Kat, Roy Jevnisek, and Shai Avidan. Matching pixels using co-occurrence statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1759, 2018.

[25] Anthony J Kearsley, Richard A Tapia, and Michael W Trosset. The solution of the metric stress and sstress problems in multidimensional scaling using newton's method. Technical report, 1994.

[26] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein. Madmm: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*, pages 680–696. Springer, 2016.

[27] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[28] Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[29] Jason Lawrence, Sean Arietta, Michael Kazhdan, Daniel Lepage, and Colleen O'Hagan. A user-assisted approach to visualizing multidimensional images. *IEEE transactions on visualization and computer graphics*, 17(10):1487–1498, 2011.

[30] Max Mignotte. Mds-based segmentation model for the fusion of contour and texture cues in natural images. *Computer Vision and Image Understanding*, 116(9):981–990, 2012.

[31] Christopher Olah. Visualizing mnist: An exploration of dimensionality reduction, 2014.

[32] Gautam Pai, Ronen Talmon, Alex Bronstein, and Ron Kimmel. Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 819–828. IEEE, 2019.

[33] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993.

[34] Guy Rosman, Alexander M Bronstein, Michael M Bronstein, Avram Sidi, and Ron Kimmel. Fast multidimensional scaling using vector extrapolation. Technical report, Computer Science Department, Technion, 2008.

[35] Gil Shamai, Yonathan Aflalo, Michael Zibulevsky, and Ron Kimmel. Classical scaling revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2255–2263, 2015.

[36] Gil Shamai and Ron Kimmel. Geodesic distance descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6410–6418, 2017.

[37] Gil Shamai, Michael Zibulevsky, and Ron Kimmel. Accelerating the computation of canonical forms for 3d nonrigid objects using multidimensional scaling. In *Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval*, pages 71–78. Eurographics Association, 2015.

[38] Gil Shamai, Michael Zibulevsky, and Ron Kimmel. Fast classical scaling. *arXiv preprint arXiv:1611.07356*, 2016.

[39] Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[40] Avram Sidi. *Vector extrapolation methods with applications*, volume 17. SIAM, 2017.

[41] Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[42] Forrest W Young. *Multidimensional scaling: History, theory, and applications*. Psychology Press, 2013.

[43] Gil Zigelman, Ron Kimmel, and Nahum Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.

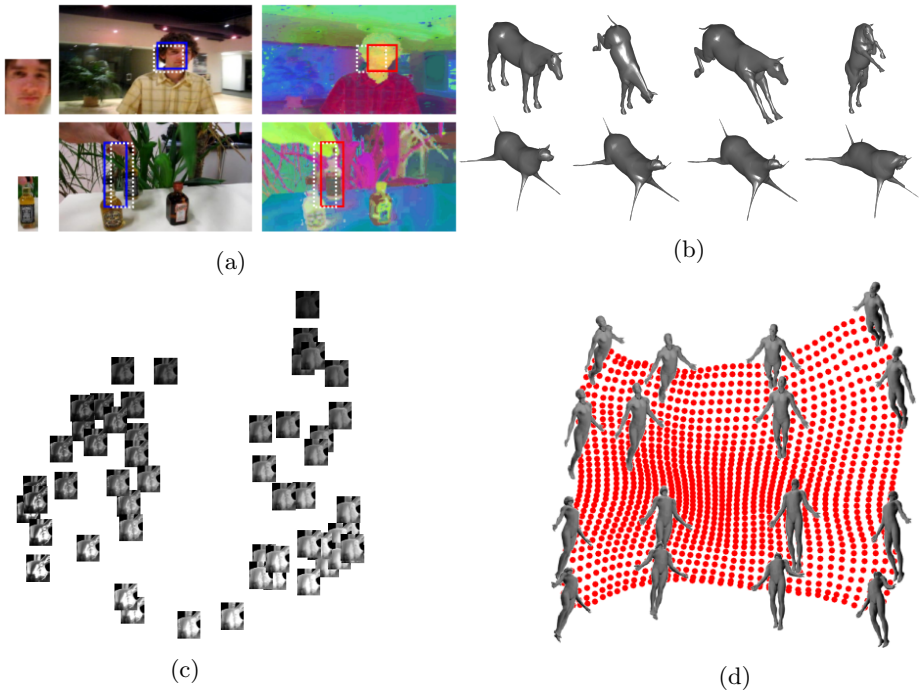

Figure 4: Applications of MDS in computer vision. (a) Template matching. The template on the left is found in the image on the right with Lucas-Kanade optical flow. Using an embedded image of co-occurence scores (right) outperforms the matching obtained with the regular RGB image (left). Image© taken from [24] with permission of the authors. (b) Canonical forms. Embedding the geodesic metric computed on the four approximately isometric horses on top, results in canonical representations of these shapes (bottom), related to each other by a rigid transformation [20, 35, 14]. The horse shape is taken from [2]. (c) Perceptual embedding of BRDF maps (i.e., governing the reflectance properties of a surface) using generalized non-metric MDS. Image© taken from [7] with permission of the authors. (d). A 3-dimensional embedding of the *camera pose manifold* (images of the same object taken with different elevation and azimuth) from a small subset of distances using a machine learning based non-linear parametric MDS model [32]. Image courtesy of the authors.