HAL
open science

Region-based Correspondence Between 3D Shapes via Spatially Smooth Biclustering

Matteo Denitto, Simone Melzi, Manuele Bicego, Umberto Castellani, Alessandro Farinelli, Mário a T Figueiredo, Yanir Kleiman, Maks Ovsjanikov

To cite this version:

Matteo Denitto, Simone Melzi, Manuele Bicego, Umberto Castellani, Alessandro Farinelli, et al.. Region-based Correspondence Between 3D Shapes via Spatially Smooth Biclustering. International Conference on Computer Vision, Oct 2017, Venice, Italy. hal-01741765

HAL Id: hal-01741765

https://hal.science/hal-01741765v1

Submitted on 23 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Region-based Correspondence Between 3D Shapes via Spatially Smooth Biclustering

Matteo Denitto[1] Simone Melzi[1] Manuele Bicego[1] Umberto Castellani[1]
matteo.denitto@univr.it
Alessandro Farinelli[1] Mário A. T. Figueiredo[2] Yanir Kleiman[3] Maks Ovsjanikov[3]

University of Verona[1] Verona, Italy
Instituto de Telecomunicações[2], IST, Univ. Lisboa, Portugal
LIX, Ecole Polytechnique[3] UMR CNRS, Palaiseau, France

Abstract

*Region-based correspondence (RBC) is a highly relevant and non-trivial computer vision problem. Given two 3D shapes, RBC seeks segments/regions on these shapes that can be reliably put in correspondence. The problem thus consists both in finding the regions and determining the correspondences between them. This problem statement is similar to that of "biclustering", implying that RBC can be cast as a biclustering problem. Here, we exploit this implication by tackling RBC via a novel biclustering approach, called S^4B (**s**patially **s**mooth **s**pike and **s**lab **b**iclustering), which: (i) casts the problem in a probabilistic low-rank matrix factorization perspective; (ii) uses a spike and slab prior to induce sparsity; (iii) is enriched with a spatial smoothness prior, based on geodesic distances, encouraging nearby vertices to belong to the same bicluster. This type of spatial prior cannot be used in classical biclustering techniques. We test the proposed approach on the FAUST dataset, outperforming both state-of-the-art RBC techniques and classical biclustering methods.*

1. Introduction

Computing correspondences between 3D shapes is a fundamental problems in computer vision, with applications in computer graphics, geometry processing, statistical shape analysis, and many others [43]. The goal of most shape matching techniques is to find correspondences between points or regions in a given pair of shapes, which can differ by a variety of deformations. This includes both near-rigid correspondences, if the shapes are related by a rotation and a translation, and the more general and challenging non-rigid shape matching problem [42], in which shapes can undergo other transformations, such as articulated motion of humans. Although several methods have been proposed to address the latter problem, most techniques either use a prescribed deformation model (*e.g.*, near-isometries or conformal deformations [3, 17]), or rely on user-provided landmark correspondences [1]. Thus, despite a significant amount of effort, the problem of finding non-rigid matches between general shapes remains challengingly open.

Another set of techniques, recently introduced in a variety of settings, relaxes the correspondence problem into that of looking for matches between probability distributions (as done in *soft maps* [38], or general real-valued functions (*e.g.*, using the functional maps framework [29]). These techniques are better suited to the general shape correspondence problem, as they do not seek precise (*e.g.*, bijective) maps between points, easily accommodating significant sampling, geometric, or even topological changes [35]. Furthermore, those soft maps can also be used as input to more refined point-based correspondence methods, to help improve the robustness and accuracy thereof [10].

One category of such generalized matching formulations is *region-based correspondence* (RBC), where the problem is formulated as that of finding regions on the shapes that behave similarly and can thus be easily put in correspondence. This problem is different from another well-studied task, called *shape co-segmentation*, since in RBC the goal is **not** to find meaningful semantic segments in various shapes (*e.g.*, limbs in animal shapes), but rather to determine regions in the two shapes that are in correspondence. As recently mentioned in [10], RBC is closely related to *biclustering* [21], in particular if the points on the shapes can be endowed with a similarity measure (*e.g.*, based on on some descriptor). Given such a measure, a similarity matrix may be built and the goal of biclustering is to simultaneously cluster both the rows and columns of this matrix [21].

Given a data matrix, biclustering aims at retrieving submatrices (*i.e.*, *biclusters*), in each of which a certain subset

of rows exhibits a "coherent behaviour" (in some sense) in a certain subset of columns. Biclustering differs from standard clustering, which treats whole rows/columns, in that the former is able to focus on local information in portions of rows and portions of columns. Many biclustering methods have been proposed, differing in the type of biclusters that can be retrieved, as well as in the adopted criteria and algorithms, as comprehensively reviewed in [21, 28].

Biclustering is a well-studied problem, with applications to gene expression data, recommender systems, market segmentation, and other areas [7, 8, 14, 16, 25, 36]. However, maybe surprisingly, biclustering has not been used for shape correspondence, with the notable exception of [10]. This is arguably due to the two following challenges: Typical 3D shapes in computer vision and graphics contain considerable *geometric* information, which is typically not taken into account in biclustering methods. Perhaps more fundamentally, in the context of RBC, an important property is required of the retrieved bi-clusters: spatial coherence, *i.e.*, nearby points should be grouped to nearby regions. This property is neither present, nor is easy to encourage, by using standard biclustering methods.

In this paper, we propose a novel biclustering technique, tailored to address the RBC problem, by explicitly encouraging spatial coherence in the corresponding regions of the given shapes. For this, we enrich the biclustering model with a spatial smoothness prior, based on geodesic distances, to encourage nearby vertices to belong to the same bicluster. This type of prior has not been considered in classical biclustering techniques, since most types of data (*e.g.*, recommender systems, gene expression data) lack spatial information. We test the proposed approach on several benchmark data, outperforming both state-of-the-art RBC techniques and classical biclustering methods.

2. Related Work and Proposed Method

Both shape correspondence and biclustering are vast areas, with comprehensive reviews being beyond the scope of this paper. Below we mention the work that we believe to be most directly related to ours, including methods for generalized (not necessarily point-to-point) correspondence and classical biclustering techniques based on probabilistic matrix factorization. For deeper discussions of both areas, the interested reader is referred to [21, 25, 43].

2.1. Generalized Shape Correspondence

As mentioned above, a recent trend in shape matching consists in finding soft correspondences between shapes. This includes methods for computing mappings between probability densities [23, 38, 39], often using formal and computational tools from optimal transport theory and measure coupling. The correspondences between probability density functions produced by these methods can, in some cases, be further refined to obtain point-wise maps [37, 40].

Another related set of techniques adopts the functional map framework, introduced in [29] and later extended significantly, *e.g.*, in [20, 32, 35]. These methods work by establishing linear mappings between general real-valued functions and have been used to find related regions on shapes (see [30] for an overview). However, most of these approaches are based on pre-segmenting the shapes and either transferring these segmentations or establishing correspondences between them as part of the pipeline (*e.g.*, [32]). Our method, in contrast, determines the optimal regions simultaneously with the correspondences.

In this area, our work is most closely related to [10], where it was recently proposed to cast RBC as a biclustering problem. That work also introduced an efficient biclustering algorithm based on a power iteration scheme. However, their method does not enforce/encourage spatial coherence on the biclusters, thus a single "cluster" can contain multiple disconnected shape regions. Our method, on the other hand, explicitly encourages spatial consistency, yielding significantly improved results, as shown below.

2.2. Biclustering

The biclustering literature includes a wide range of approaches, comprehensively reviewed in [21, 28]. A recent trend is to use *matrix factorization* tools [4, 11, 13, 19, 31, 44], most of which relying on the concept of *latent block models* [11, 26, 31, 34, 44]. These approaches simultaneously arrange the rows and columns of a data matrix into groups of similar response patterns, thus yielding biclusters where the rows/columns belong to only one group, and the data matrix is divided into exhaustive and non overlapping biclusters. These characteristic can limit the applicability of these methods in some contexts. Moreover, most biclustering formulations and methods are invariant under permutations of the rows/columns, *i.e.*, they treat them as independent entities, ignoring any neighbourhood relationship (spatial or otherwise) between rows/columns of the matrix.

2.3. Overview of the Proposed Method

This paper introduces a novel algorithm, where the rationale is to decompose the data matrix into levels, each corresponding to a different bicluster (as in [13, 4]), thus allowing to obtain non-exhaustive and possibly overlapping biclusters. In this class of approaches, sparsity plays a crucial role. In fact, the data matrices to which biclustering is typically applied have large numbers of rows and columns (*e.g.*, thousands by hundreds, in gene expression data), but the biclusters often involve only small portions thereof. The proposed method results from combining a probabilistic low-rank matrix factorization criterion with a spike and slab prior to encourage sparsity. Spike and slab priors were pro-

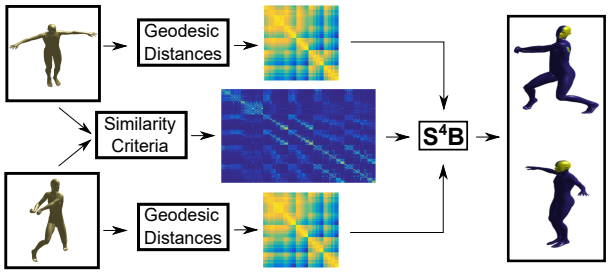


Figure 1: The framework of the the **s**patially **s**mooth **s**pike and **s**lab **b**iclustering (S^4B) for the RBC problem

posed in [24] for variable selection in linear regression, and later generalized and adopted by many authors as general-purpose sparsity-inducing priors [15]. The other main novel ingredient herein proposed is a spatial smoothness prior, to improve the bicluster coherence/quality. Similarly to what has been proposed for clustering [6], we exploit known pair-wise relations to encourage certain rows (and/or columns) to belong to the same bicluster. This is obtained by combining the spike and slab prior with two pair-wise priors, one for the rows and one for the columns, encouraging pairs of nearby rows/columns to be grouped together. We call our method **s**patially **s**mooth **s**pike and **s**lab **b**iclustering (S^4B).

The spatial priors are derived from two similarity matrices (one for the rows and one for the columns), based on the geodesic distances between the shapes' vertices. Those matrices control the strength with which each pair of vertices (on each shape) is encouraged to belong to the same bicluster. Consequently, the proposed S^4B method combining two types of information (as depicted in Fig. 1): (i) the affinity between pairs of vertices of different shapes; (ii) the neighbourliness between vertices on the same shape.

Notation: Before continuing, it is convenient to define some notation. Matrices are denoted by capital letters (*e.g.*, D, V, Z), vectors with lower-case letters (*e.g.*, d, v, z), and matrix/vector elements by using subscripts (*e.g.*, the entry (i, j) of matrix D is d_{ij} and the component p of vector d is d_p). The so-called vec (vectorization) operator produces a vector with the elements of its matrix argument stacked column-by-column; its inverse is denoted vec^{-1}. A pair of useful equalities concerning the vec operator are

$$\text{vec}(AB) = (I \otimes A)\text{vec}(B) = (B^T \otimes I)\text{vec}(A), \quad (1)$$

where I is an identity matrix of adequate dimensions and $\otimes$ is the Kronecker matrix product [22]. Finally, given some matrix A, $\|A\|_F$ denotes its Frobenius norm, which is the Euclidean norm of its vectorization: $\|A\|_F = \|\text{vec}(A)\|_2$.

3. Biclustering via Sparse Low-Rank Factorization

Let $D \in \mathbb{R}^{n\times m}$ be the given data matrix, and $R = \{1, \ldots, n\}$ and $C = \{1, \ldots, m\}$ be the sets of row and column indices, respectively. Let D_{TK}, where $T \subseteq R$ and $K \subseteq C$, be the submatrix with the subset of rows in T and the subset of columns in K. With this notation, a *bicluster* is a submatrix D_{TK}, such that the subset of rows of D with indices in T exhibits a "coherent behavior" (in some sense) across the set of columns with indices in K, and vice versa.

The choice of *coherence* criterion controls the type of biclusters to be retrieved [21, 28]. A standard choice is that entries in the same bicluster have similar values, significantly different from the other entries of the matrix. For example, a data matrix containing one bicluster with rows $T = \{1, 2\}$ and columns $K = \{1, 2\}$ may look like

$$D = \begin{bmatrix} 10 & 10 & 0 & 0 \\ 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

From an algebraic point of view, this matrix can be represented by the outer product $D = vz^T$ of the vectors

$$v = [5,\ 5,\ 0,\ 0,\ 0]^T \quad \text{and} \quad z = [2,\ 2,\ 0,\ 0]^T.$$

Generalizing to k biclusters, we can formulate the problem as the decomposition of a given data matrix D as the sum of k outer products,

$$D = \sum_{i=1}^{k} v_i z_i^T = \underbrace{[v_1, ..., v_k]}_{V\in\mathbb{R}^{n\times k}} \underbrace{[z_1, ..., z_k]^T}_{Z\in\mathbb{R}^{k\times m}} = VZ. \quad (2)$$

The link between biclustering and sparse low-rank matrix factorization is clear: (i) the product VZ has rank no larger than k (the number of biclusters); (ii) if D is much larger than the expected biclusters (as it is typically the case), each v_i and z_i should be sparse, thus V and Z should be sparse.

4. The S^{4}B Approach

This section provides a formal explanation of how the three main ingredients of S^4B are formulated:

(i) low-rank matrix factorization;

(ii) factor sparsity;

(iii) spatial smoothness.

4.1. Spike and Slab

The spike and slab is a univariate sparsity-inducing prior composed by the mixture of two zero-mean Gaussian distributions: one with very small variance, modeling a high probability of nearly zero values; another one with large variance, modeling the presence of large values. Under this

density, both very large and very small (nearly zero) samples have high likelihood, which is impossible under a single Gaussian. The idea is that by sampling from the low-variance Gaussian yields background values, whereas sampling from the high-variance Gaussian produces foreground values. Formally, the spike and slab prior has the form

$$\mathcal{P}(x|\alpha,\tau_1,\tau_2) = \alpha\,\mathcal{N}(x|0,\tau_1^2) + (1-\alpha)\mathcal{N}(x|0,\tau_2^2), \quad (3)$$

with $\tau_2 \ll \tau_1$, parameter $\alpha \in [0,\,1]$ regulates the sparsity degree, and $\mathcal{N}(x|\mu,\sigma^2)$ denotes a Gaussian density with mean μ and variance σ^2, computed at x. Note that (3) is equivalent to the following two-stage model

$$\mathcal{P}(x|h,\tau_1,\tau_2) = \mathcal{N}(x|0,\tau_1^2)^h\mathcal{N}(x|0,\tau_2^2)^{(1-h)}, \quad (4)$$

$$\mathcal{P}(h|\alpha) = \alpha^h(1-\alpha)^{1-h}, \quad (5)$$

where $h \in \{0,1\}$ is a latent binary variable following a Bernoulli distribution of parameter α. The mixture in (3) results from marginalizing this model with respect to h.

4.2. The S^{4}B Model

The data matrix $D \in \mathbb{R}^{n\times m}$ is modeled as a noisy version of a low-rank product VZ,

$$\begin{aligned}\mathcal{P}(D|V,Z,\sigma^2) &= \mathcal{N}(D|VZ,\sigma^2 I) \quad (6)\\ &= \prod_{i=1}^{n}\prod_{j=1}^{m}\mathcal{N}(d_{ij}|(VZ)_{ij},\sigma^2), \quad (7)\end{aligned}$$

where $V \in \mathbb{R}^{n\times k}$, $Z \in \mathbb{R}^{k\times m}$, k is the number of biclusters, and σ is the standard deviation of the noise, which also accounts for approximation errors. This part embodies the low-rank assumption, as VZ has rank no larger than k.

As explained above, matrices V and Z are expected to be *sparse*, to keep the biclusters of small size. Moreover, we would like to enforce certain pairs of rows/columns to be in the same bicluster. A hierarchical prior expressing these desiderata is formally defined as follows (see Figure 2):

- The entries of V follow a *pairwise regularized* spike and slab prior with variances τ_1^2 and τ_2^2 (such that $\tau_1^2 \gg \tau_2^2$),

$$\begin{aligned}\mathcal{P}(V|H,\tau_1,\tau_2) &= \frac{1}{\Xi}\left[\prod_{j=1}^{k}\exp\left(-\frac{\beta}{4}\sum_{i=1}^{n}\sum_{l=1}^{n}S^v_{il}(v_{ij}-v_{lj})^2\right)\right]\\ &\quad\left(\prod_{i=1}^{n}\prod_{j=1}^{k}\mathcal{N}(v_{ij}|0,\tau_1^2)^{h_{ij}}\mathcal{N}(v_{ij}|0,\tau_2^2)^{1-h_{ij}}\right)\\ &= \frac{1}{\Xi}\left[\prod_{j=1}^{k}\exp\left(-\frac{1}{2}v_j^T\Delta_v v_j\right)\right]\\ &\quad\left(\prod_{i=1}^{n}\prod_{j=1}^{k}\mathcal{N}(v_{ij}|0,\tau_1^2)^{h_{ij}}\mathcal{N}(v_{ij}|0,\tau_2^2)^{1-h_{ij}}\right)\end{aligned} \quad (8)$$

  where v_j is the j-th column of V and $S^v_{il} = S^v_{li} \geq 0$ is the strength with which v_{ij} and v_{lj} are encouraged to be similar (*i.e.*, in the same bicluster); Δ_v is the $n \times n$ Laplacian of a graph with edge weights S^v_{il},

$$\Delta_v = \beta\left(\text{diag}\left(\sum_{i=1}^{n}S^v_{1i},\ldots,\sum_{i=1}^{n}S^v_{ni}\right) - S^v\right), \quad (9)$$

  and S^v is the matrix with elements S^v_{il}; parameter β controls the global weight of the pair-wise (smoothness) prior; each h_{ij} is an entry of the matrix H of latent variables of the spike-and-slab prior; finally, Ξ is the normalization constant, which does not need to be known.

- The binary latent variables in H are independent and follow a Bernoulli distribution of parameter α_1,

$$\mathcal{P}(H|\alpha_1) = \prod_{i=1}^{n}\prod_{j=1}^{k}\alpha_1^{h_{ij}}(1-\alpha_1)^{1-h_{ij}}. \quad (10)$$

- The entries of Z follow a *pairwise regularized* spike and slab prior, with variances ρ_1^2 and ρ_2^2 (such that $\rho_1^2 \gg \rho_2^2$),

$$\begin{aligned}\mathcal{P}(Z|G,\rho_1,\rho_2) &= \frac{1}{\Xi}\left[\prod_{j=1}^{k}\exp\left(-\frac{\beta}{4}\sum_{i=1}^{m}\sum_{l=1}^{m}S^z_{il}(z_{ij}-z_{lj})^2\right)\right]\\ &\quad\left(\prod_{i=1}^{k}\prod_{j=1}^{m}\mathcal{N}(z_{ij}|0,\rho_1^2)^{g_{ij}}\mathcal{N}(z_{ij}|0,\rho_2^2)^{1-g_{ij}}\right)\\ &= \frac{1}{\Xi}\left[\prod_{j=1}^{k}\exp\left(-\frac{1}{2}z_j^T\Delta_z z_j\right)\right]\\ &\quad\left(\prod_{i=1}^{k}\prod_{j=1}^{m}\mathcal{N}(z_{ij}|0,\rho_1^2)^{g_{ij}}\mathcal{N}(z_{ij}|0,\rho_2^2)^{1-g_{ij}}\right)\end{aligned} \quad (11)$$

  where vector z_j is the j-th column of Z and $S^z_{il} = S^z_{li} \geq 0$ is the strength with which z_{ij} and z_{lj} are encouraged to be similar (*i.e.*, in the same bicluster); Δ_z is the $m \times m$ Laplacian of a graph with edge weights S^z_{il},

$$\Delta_z = \beta\left(\text{diag}\left(\sum_{i=1}^{m}S^z_{1i},\ldots,\sum_{i=1}^{m}S^z_{mi}\right) - S^z\right), \quad (12)$$

  S^z is the matrix with elements S^z_{il}.

- The binary latent variables in matrix G follow a Bernoulli distribution of parameter α_2,

$$\mathcal{P}(G|\alpha_2) = \prod_{i=1}^{k}\prod_{j=1}^{m}\alpha_2^{g_{ij}}(1-\alpha_1)^{1-g_{ij}}. \quad (13)$$

Intuitively, α_1 and α_2 control the sparsity of V and Z, *i.e.*, the biclusters sizes on rows and columns, respectively. The variances $\tau_1^2, \tau_2^2, \rho_1^2$, and ρ_2^2 control the value ranges.

The joint distribution of all the variables and parameters involved in this model can now be written as

$$\begin{aligned}&\mathcal{P}(D,V,Z,H,G,\sigma,\tau_1,\tau_2,\rho_1,\rho_2,\alpha_1,\alpha_2)\\ &= \mathcal{P}(D|V,Z,\sigma^2)\mathcal{P}(V|H,\tau_1,\tau_2)\mathcal{P}(Z|G,\rho_1,\rho_2)\\ &\quad\mathcal{P}(H|\alpha_1)\mathcal{P}(G|\alpha_2)\mathcal{P}(\sigma,\tau_1,\tau_2,\rho_1,\rho_2,\alpha_1,\alpha_2), \quad (14)\end{aligned}$$

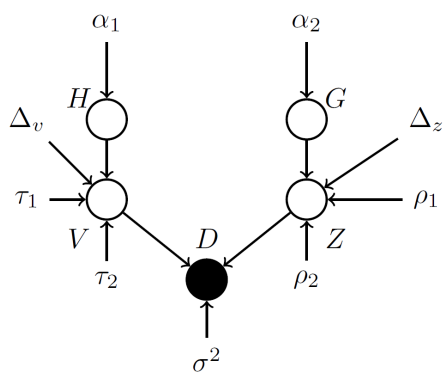


Figure 2: Graph of the probabilistic model in Section 4.2.

where $\mathcal{P}(\tau_1, \tau_2, \rho_1, \rho_2, \alpha_1, \alpha_2)$ is a prior on the model parameters. In this paper, we consider this prior to be flat, that is, we seek *maximum likelihood* (ML) estimates thereof.

Finally, this model may be easily extended to the case where each bicluster has its own parameter set (spike and slab variances and mixing probability), rather than being assumed the same for all the biclusters. In this paper, we keep the simpler version that we have just introduced.

4.3. Parameter Estimation

We estimate the model parameters $\tau_1, \tau_2, \rho_1, \rho_2, \alpha_1, \alpha_2$, and σ by resorting to the classical EM algorithm [9]. Recall that EM is a class of iterative algorithms designed to obtain marginal ML estimates $\hat{\theta} = \arg\max_\theta \mathcal{P}(x|\theta)$, where the marginal likelihood results from marginalizing out hidden/latent variables y, *i.e.*, $\mathcal{P}(x|\theta) = \int \mathcal{P}(x, y|\theta)\, dy$ (with summation rather than integration, if y is discrete). The algorithm alternates between two steps:

E-step: computes the conditional expectation of the complete log-likelihood, given the current parameter estimate $\hat{\theta}^{(t)}$ and the observed data x, yielding the so-called Q-function:

$$Q(\theta, \hat{\theta}^{(t)}) = \mathbb{E}_y \left[\log \mathcal{P}(x, y|\theta) | x, \hat{\theta}^{(t)} \right]. \tag{15}$$

M-step: updates the parameter estimate according to

$$\hat{\theta}^{(t+1)} = \arg\max_\theta Q(\theta, \hat{\theta}^{(t)}). \tag{16}$$

Computing the expectation in (15) may not be trivial in general, as it may involve intractable integration.

Concerning the unobserved V, Z, H, and G, there is a choice of marginalizing them out via the EM algorithm, *i.e.*, treat them as latent variables, or maximizing with respect to them, *i.e.* seeing them as parameters, rather than latent variables. Inspired by [6], and in order to obtain a simpler E-Step, we treat H and G as latent variables, but V and Z as unknown parameters, estimated along with $\tau_1, \tau_2, \rho_1, \rho_2, \alpha_1, \alpha_2$, and σ. We could also treat H and G as parameters; however, as these are binary matrices, maximizing with respect to them corresponds to making hard decisions, which may have a strong influence in the whole optimization procedure. On the other hand, V and Z are matrices of real-valued entries, thus estimating them has a smoother/weaker influence on the other estimates. For these reasons, V and Z are treated as parameters, and H and G as latent variables. Next, we present the form that the E-step and the M-step take in the proposed S^4B model.

4.3.1 The E-Step

To keep the notation compact, we denote the complete tuple of parameters as $\theta = (V, Z, \sigma^2, \alpha_1, \alpha_2, \tau_1, \tau_2, \rho_1, \rho_2)$. Recall that the joint distribution of all the variables and parameters is as given in (14). With D observed and H and G latent, the Q-function is obtained by computing

$$\mathcal{Q}(\theta, \hat{\theta}^{(t)}) = \mathbb{E}_{H,G} \Big[\log \mathcal{P}(D, H, G, \theta) \big| \hat{\theta}^{(t)}, D \Big].$$

Straightforward, but long and tedious, manipulations and dropping terms that do not depend on θ, leads to

$$\begin{aligned} &\mathcal{Q}(\theta, \hat{\theta}^{(t)}) = \\ &- \frac{nm}{2} \log(\sigma^2) - \frac{||D - VZ||^2}{2\sigma^2} \\ &- \frac{||\overline{H}^{(t)}||_F}{2} \log(\tau_1^2) - \frac{||1 - \overline{H}^{(t)}||_F}{2} \log(\tau_2^2) \\ &- \frac{||\overline{G}^{(t)}||_F}{2} \log(\rho_1^2) - \frac{||1 - \overline{G}^{(t)}||_F}{2} \log(\rho_2^2) \\ &- \frac{1}{2} v^T \overline{H}^{(t)} v - \frac{1}{2} z^T \overline{G}^{(t)} z \\ &+ \left(\sum_{p=1}^{nk} \overline{h}_p^{(t)} \right) \log\Big(\frac{\alpha_1}{1 - \alpha_1} \Big) + n\, k \log(1 - \alpha_1) \\ &+ \left(\sum_{j=1}^{km} \overline{g}_j^{(t)} \right) \log\Big(\frac{\alpha_2}{1 - \alpha_2} \Big) + k\, m \log(1 - \alpha_2) \end{aligned} \tag{17}$$

where $v = \text{vec}(V)$, $z = \text{vec}(Z)$,

$$\overline{H}^{(t)} = \text{diag}\Big(\frac{\overline{h}_1^{(t)}}{\tau_1^2} + \frac{1 - \overline{h}_1^{(t)}}{\tau_2^2}, ..., \frac{\overline{h}_{nk}^{(t)}}{\tau_1^2} + \frac{1 - \overline{h}_{nk}^{(t)}}{\tau_2^2} \Big), \tag{18}$$

$$\overline{G}^{(t)} = \text{diag}\Big(\frac{\overline{g}_1^{(t)}}{\rho_1^2} + \frac{1 - \overline{g}_1^{(t)}}{\rho_2^2}, ..., \frac{\overline{g}_{km}^{(t)}}{\rho_1^2} + \frac{1 - \overline{h}_{km}^{(t)}}{\rho_2^2} \Big), \tag{19}$$

and, for $p = 1, ..., nk$, and $j = 1, ..., km$,

$$\overline{h}_p^{(t)} = \frac{\alpha_1 \mathcal{N}(v_p|0, \tau_1^2)}{\alpha_1 \mathcal{N}(v_p|0, \tau_1^2) + (1 - \alpha_1) \mathcal{N}(v_p|0, \tau_2^2)} \tag{20}$$

$$\overline{g}_j^{(t)} = \frac{\alpha_2 \mathcal{N}(z_j|0, \rho_1^2)}{\alpha_2 \mathcal{N}(z_j|0, \rho_1^2) + (1 - \alpha_2) \mathcal{N}(z_j|0, \rho_2^2)}. \tag{21}$$

4.3.2 The M-Step

Examining the several terms in (17) reveals the following: maximizing with respect to V and Z corresponds to a variant of a low-rank matrix factorization formulation [5]; concerning the other parameters, closed-form updates can be obtained by equating the corresponding derivatives to zero.

Updating V and Z: considering only the terms in (17) that depend on V and Z, we have the following low-rank factorization problem,

$$\min_{V,Z}\Big[\frac{||D-VZ||_F^2}{2\sigma^2}+\frac{1}{2}v^T(\overline{H}^{(t)}+\overline{\Delta_v})v+\frac{1}{2}z^T(\overline{G}^{(t)}+\overline{\Delta_z})z\Big], \tag{22}$$

where $\overline{\Delta_v}$ = block-diag$(\Delta_v,\ldots,\Delta_v)$ is an $(nk)\times(nk)$ block diagonal matrix with k copies of the Δ_v (similarly for $\overline{\Delta_z}$ and Δ_z). This is a generalization of the model recently proposed in [5], where the generalization consists in replacing the Frobenius norm used in [5] by the quadratic terms $v^T(\overline{H}^{(t)}+\overline{\Delta_v})v$ and $z^T(\overline{G}^{(t)}+\overline{\Delta_z})z$. In the absence of matrices $\overline{\Delta_v}$ and $\overline{\Delta_z}$, these would simply be squared weighted Frobenius norms.

Following [5], we tackle (22) via the *augmented Lagrangian method* (ALM), also known as *method of multipliers* [12, 33]. The first step is to re-write (22) as an equivalent constrained problem, by introducing a new variable C to take the place of the low rank product VZ,

$$\min_{V,Z,C}\Big[\frac{||D-C||_F^2}{2\sigma^2}+\frac{1}{2}v^T\widetilde{H}v+\frac{1}{2}z^T\widetilde{G}z\Big] \tag{23}$$
$$\text{s.t.}\quad C=VZ,$$

where, to keep the notation lighter, we used $\widetilde{H}=\overline{H}^{(t)}+\overline{\Delta_v}$ and $\widetilde{G}=\overline{G}^{(t)}+\overline{\Delta_z}$. It is convenient to write a fully vectorized version of this problem; to that end (and as for $v=\text{vec}(V)$ and $z=\text{vec}(Z)$), we define $c=\text{vec}(C)$ and $d=\text{vec}(D)$, leading to

$$\min_{v,z,c}\Big[\frac{||d-c||_2^2}{2\sigma^2}+\frac{1}{2}v^T\widetilde{H}v+\frac{1}{2}z^T\widetilde{G}z\Big] \tag{24}$$
$$\text{s.t.}\quad c=(I\otimes V)z,$$

where the constraint $c=(I\otimes V)z$ is equivalent to $C=VZ$ (as is clear from (1)). Notice that the constraint can also be written as $c=(Z^T\otimes I)v$ (as is also clear from (1)). For later use, we define the two following matrices:

$$A(z)=(Z^T\otimes I)\qquad\text{and}\qquad B(v)=(I\otimes V). \tag{25}$$

The augmented Lagrangian is obtained by adding a quadratic penalty to the Lagrange function of (24),

$$\mathcal{L}_\rho(v,z,c,y) = \frac{||d-c||^2}{2\sigma^2}+\frac{1}{2}v^T\overline{H}v+\frac{1}{2}z^T\overline{G}z+ \frac{\rho}{2}||B(v)z-c||^2+y^T\big(c-B(v)z\big), \tag{26}$$

where y is the vector of Lagrange multipliers, $\rho\geq 0$ is a parameter. ALM proceeds by alternating between minimizing $\mathcal{L}_\rho(v,z,c,y)$ with respect to v,z,c, and updating the Lagrange multipliers y.

Unfortunately, $\mathcal{L}_\rho(v,z,c,y)$ cannot be minimized in closed-form simultaneously with respect to v,z,c, thus we follow the approach in [5] and solve it by a *non-linear block Gauss-Seidel* (NLBGS) method, *i.e.*, by cycling through minimizations w.r.t. v,z, and c, until some convergence criterion is satisfied, taking advantage of the fact that each of these minimizations can be written in closed form. Letting the iteration counter of the NLBGS be s and denoting $A_s=A(z^{(s)})$ and $B_s=B(v^{(s)})$, the resulting update expressions are (for $s=1,2,\ldots$)

$$v^{(s+1)}=\Big(\widetilde{H}+\rho A_s^TA_s\Big)^{-1}\Big(A_s^Ty+\rho A_s^Tc^{(s)}\Big) \tag{27}$$

$$z^{(s+1)}=\Big(\widetilde{G}+\rho B_{s+1}^TB_{s+1}\Big)^{-1}\Big(\big(B_{s+1}^Ty+\rho B_{s+1}^Tc^{(s)}\Big) \tag{28}$$

$$c^{(s+1)}=\frac{d-\sigma^2y+\rho B_{s+1}z^{(s+1)}}{1+\sigma^2\rho}. \tag{29}$$

Summarizing, $V^{(t+1)}$ and $Z^{(t+1)}$, which are the solution of (22), are obtained by cycling through (27), (28), and (29), until some convergence criterion is satisfied.

Other parameters: the updates of other parameters $(\tau_1^2,\tau_2^2,\rho_1^2,\rho_2^2,\sigma,\alpha_1,\alpha_2)$ are obtained by setting the corresponding partial derivatives of $\mathcal{Q}(\theta,\hat{\theta}^{(t)})$ to zero, yielding

$$\tau_1^2=\big(v^T\overline{H}^{(t)}v\big)/\|\overline{H}^{(t)}\|_F \tag{30}$$

$$\tau_2^2=v^T(1-\overline{H}^{(t)})v/\|1-\overline{H}^{(t)}\|_F \tag{31}$$

$$\rho_1^2=z^T\overline{G}^{(t)}z/\|\overline{G}^{(t)}\|_F \tag{32}$$

$$\rho_2^2=z^T(1-\overline{G}^{(t)})z/\|1-\overline{G}^{(t)}\|_F \tag{33}$$

$$\alpha_1=\Big(\sum_{p=1}^{nk}\overline{h}_p\Big)/(nk) \tag{34}$$

$$\alpha_2=\Big(\sum_{p=1}^{nk}\overline{g}_p\Big)/(mk) \tag{35}$$

$$\sigma^2=\|D-VZ\|_F^2/(nm), \tag{36}$$

where we have omitted the iteration counter superscript $(\cdot)^{(t)}$, to keep the notation lighter.

Complexity: the leading term concerning space complexity is $\mathcal{O}(nk)$ (or $\mathcal{O}(km)$, which is the space needed to store the $A(z)$ (or $B(v)$) matrix. Thus, an adequate sparse representation can overcome this possible drawback. Regarding time complexity (for each iteration), the leading term is $\mathcal{O}(n^3k^3)$ (or $\mathcal{O}(m^3k^3)$) which is the worst case scenario

for matrix multiplication/inversion of a $\mathcal{O}(nk)$ (or $\mathcal{O}(mk)$) matrix.

5. Experiments

This section describes the experiments carried out to compare S^4B with the relevant state-of-the-art, including specific RBC and general biclustering techniques. Regarding RBC-specific techniques, the most relevant work is the recent *stable region correspondences* (SRC) approach [10], which uses a power iteration scheme. We can directly compare the corresponding regions obtained by S^4B and SRC with respect to the ground-truth mapping between the two shapes. Other methods produce point-to-point correspondences based on geometric features of the shapes. We compare our corresponding regions to *blended intrinsic maps* (BIM) [17], a popular point-to-point correspondence method. To evaluate BIM in the context of corresponding regions, we follow [10] and use the point-to-point mapping to transport the segmentation computed on one shape to the other. We also provide a comparison between S^4B and its version without the spatial smoothness prior, to show that this prior is crucial to obtain high quality results.

All the experiments use FAUST [2], a challenging recent dataset containing 100 scanned human shapes (10 poses of 10 subjects). This dataset presents both near-isometric (different poses of the same subject) and non-isometric deformations (due to the significant variability between different subjects). All of the shapes have the same number of vertices, and the ground-truth one-to-one correspondence (or map) between each pair of shapes is available. We measure the quality of the results as the global labelling accuracy with respect to the ground-truth map. More precisely, since every method assigns a label to each vertex on the two shapes, we compute a score that sums the influence area of vertices in one shape that are given the same label as their mapping in the second shape. This provides the percentage of the shape's area that has a correct correspondence. Then, we compute the same score by inverting the role of first and second shape, and consider the mean of these two scores as the final score. Formally,

$$\text{score}(L_a, L_b) = \sum_{i=1}^{n} \Big(L_a(i) = L_b\big(f(i)\big) \Big) A_a(f(i)) \quad (37)$$

$$\text{quality} = \frac{1}{2}\Big(\frac{\text{score}(L_a, L_b)}{\sum(A_a)} + \frac{\text{score}(L_b, L_a)}{\sum(A_b)} \Big), \quad (38)$$

where a and b are the shapes, L_a and L_b are the given labels, A_a and A_b indicate the influence area of each vertex, and f is the ground-truth point-to-point mapping.

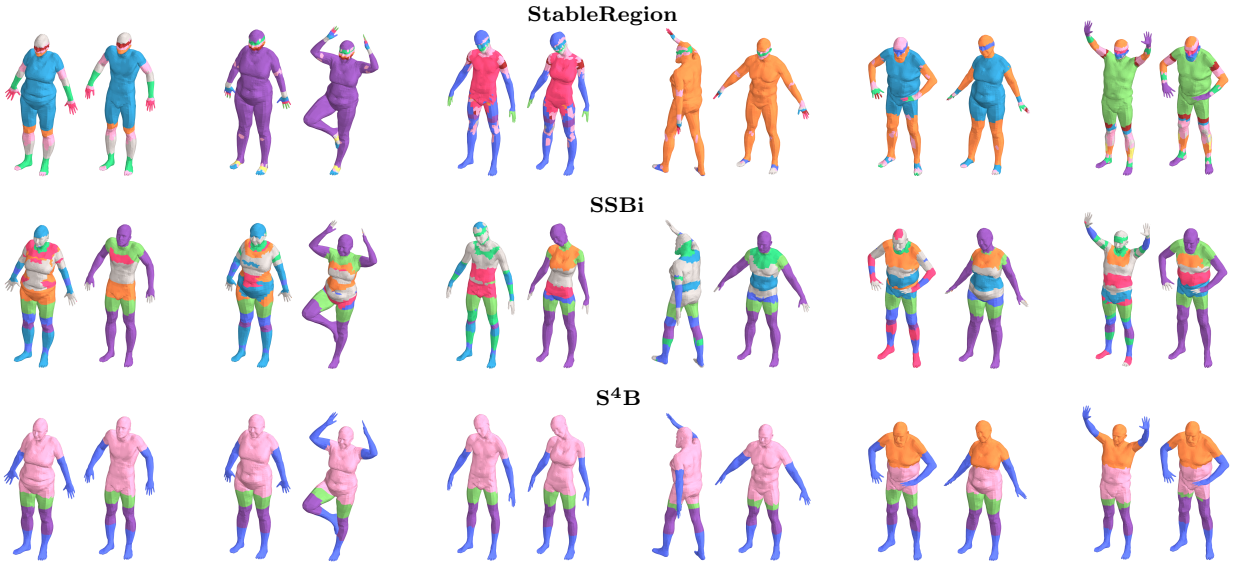
To evaluate the SRC method, we followed the guidelines provided by its authors [10]. Concerning BIM, since its performance is highly influenced by the starting segmentation, we evaluate the point-to-point mapping using two possible segmentations: (i) based on geodesic Voronoi cells around a *farthest point sampling* [41], which provides segments of uniform size; (ii) based on the output labels of S^4B . This gives us a starting segmentation, which we transfer to the second shape using the correspondences provided by BIM.

Implementation details. For S^4B , we used $k = 8$ biclusters, and set $beta$ to be three times the maximum value in the affinity matrix (computed as described in [10], with the same descriptors). For the geodesic similarities, we considered as "near" only distances below 5% of the maximum. Notice that setting $\beta = 0$ turns off spatial smoothness prior, thus obtaining a standard biclustering algorithm, hereafter referred to as *spike and slab biclustering* (SSBi). We used parameters that lead to similar numbers of segments for each of the methods tested, for fairness of comparison. Once the method converges, H and G contain the probability that each vertex belongs to the retrieved biclusters. Finally, we assign each vertex to the bicluster (and hence the label) maximizing that probability (discarding labels with probability below 0.75).

Evaluation. We randomly selected 50 pairs of shapes from the FAUST dataset and applied each of the previously mentioned methods. The results reported in Table 1 show that S^4B performs better than SRC, BIM, and SSBi. Particularly, S^4B outperforms both the SRC and BIM-Voronoi approaches. By comparing S^4B and SSBi, we can state that the spatial prior is crucial to obtain high quality results. All of these results have been statistically evaluated with a paired T-test with significance level equal to 5%.

Figure 3 shows some of the results obtained: the first row shows results by the SRC method [10]; the second row shows results of SSBi; the third row presents results of the new S^4B . Clearly, the S4B results present high coherence among the different pairs of shapes (although they have obviously been analysed independently), moreover they produce more connected regions, unlike those obtained by SRC and SSBi.

Application We can exploit the obtained segments in several applications. One of these could be point-wise map estimation through the functional maps algorithm [29, 27]. This recent framework provides a point-to-point map between a pair of shapes, starting from some vertices descriptors and some region correspondences between the pair of shapes [**?**]. We thus compare the standard baseline (without region correspondences) with the regions provided by both S^4B and SRC. Figure 4 plots the performance of S^4B and SRC, using the standard *correspondence quality characteristic* [18], as a function of the radius r, with each point in the curves representing the matching percentage considering only points that are at distant less than r from their ground truth correspondence. This results show that the performances clearly increase by adding the region indicators



082 – 092 042 – 047 005 – 065 028 – 070 021 – 045 039 – 021

Figure 3: Qualitative results of *SRC*, *SSBi* and S^4B on the FAUST dataset.

	Stable Region	BIM Voronoi	BIM S^4B	SSBi	S^4B
scenario1	95.37	95.87	**97.98**	29.91	97.36
scenario2	85.34	95.35	94.21	30.39	**95.73**
scenario3	85.39	92.51	92.5	32.32	**94.25**
global	86.58	93.26	93.36	31.8	**94.8**

	Stable Region	BIM Voronoi	BIM S^4B	SSBi	S^4B
scenario1	94.95	96.76	97.84	30.07	**97.98**
scenario2	87.42	96.17	93.82	30.95	**96.63**
scenario3	87.63	92.82	92.55	33.77	**94.96**
global	89.33	93.1	93.15	31.26	**95.52**

Table 1: Results on the FAUST dataset using *SRC*, *BIM*, *SSBi*, and the proposed S^4B . Scenario1: pairs of shapes of the same subject in different poses. Scenario2: pairs of different subjects in the same pose. Scenario3: pairs of different subjects in different poses. The left/right tables show mean/median scores for each scenario, and the global mean/median score.

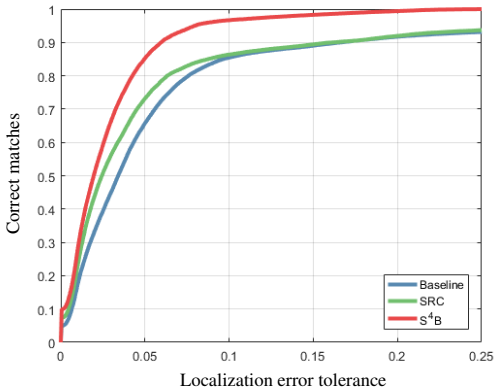


Figure 4: Performances of S^4B and SRC results when adopted to create a point-wise estimation map.

obtained from S^4B . Further, also adding region indicators from SRC improves the baseline results; however it seems that the stability of our regions are clearly more informative since they produce a stronger improvement on the quality of the functional maps.

6. Conclusions

We have presented a new approach to tackle the *region-based correspondence* (RBC) problem, where the goal is to find segments/regions on two shapes that can be reliably put in correspondence. When formulated using a given similarity matrix between vertices on the two shapes, RBC is a biclustering problem. However, in RBC, it is important that the obtained biclusters have spatial smoothness/coherence (*i.e.*, that nearby vertices are put in the same region), a desire that is not easily cast into classical biclustering techniques. We have thus proposed a novel probabilistic biclustering formulation combining three ingredients: low-rank matrix factorization; a sparsity-inducing spike and slab prior (SSP); a spatial smoothness prior coupled with the SSP, encouraging pairs of nearby (in terms of geodesic distance of the corresponding vertices) rows/columns to belong to the same bicluster. Computationally, we resort to the EM algorithm to estimate the model parameters and retrieve the biclusters. The resulting method, called *spatially smooth spike and slab biclustering* (S^4B), was tested on the FAUST dataset, outperforming both state-of-the-art RBC techniques and classical biclustering methods.

Acknowledgements

Parts of this work were supported by the Marie-Curie CIG-334283, a CNRS chaire d'excellence, chaire Jean Marjoulet from Ecole Polytechnique, FUI project TANDEM 2, and a Google Focused Research Award.

References

[1] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2016. 1

[2] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014. 7

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006. 1

[4] K. Bunte, E. Leppäaho, I. Saarinen, and S. Kaski. Sparse group factor analysis for biclustering of multiple data sources. *Bioinformatics*, 32(16):2457–2463, 2016. 2

[5] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2488–2495. IEEE, 2013. 6

[6] D. S. Cheng, V. Murino, and M. Figueiredo. Clustering under prior knowledge with application to image segmentation. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 401–408. MIT Press, 2007. 3, 5

[7] P. de Castro, F. de França, H. Ferreira, and F. Von Zuben. Applying biclustering to text mining: an immune-inspired approach. *Artificial Immune Systems*, pages 83–94, 2007. 2

[8] S. Dolnicar, S. Kaiser, K. Lazarevski, and F. Leisch. Biclustering overcoming data dimensionality problems in market segmentation. *Journal of Travel Research*, 51(1):41–49, 2012. 2

[9] B. J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(9):1392–1416, 2005. 5

[10] V. Ganapathi-Subramanian, B. Thibert, M. Ovsjanikov, and L. Guibas. Stable region correspondences between non-isometric shapes. In *Computer Graphics Forum*, volume 35, pages 121–133. Wiley Online Library, 2016. 1, 2, 7

[11] G. Govaert and M. Nadif. An em algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and machine intelligence*, 27(4):643–647, 2005. 2

[12] M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969. 6

[13] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. Van Sanden, D. Lin, W. Talloen, et al. Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010. 2

[14] A. A. Irissappane, S. Jiang, and J. Zhang. A biclustering-based approach to filter dishonest advisors in multi-criteria e-marketplaces. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1385–1386. International Foundation for Autonomous Agents and Multiagent Systems, 2014. 2

[15] H. Ishwaran and J. S. Rao. Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773, 2005. 3

[16] M. Kaytoue, V. Codocedo, A. Buzmakov, J. Baixeries, S. O. Kuznetsov, and A. Napoli. Pattern structures and concept lattices for data mining and knowledge processing. In *Machine Learning and Knowledge Discovery in Databases*, pages 227–231. Springer, 2015. 2

[17] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 1, 7

[18] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *ACM Transaction on Graphics (TOG)*, 30(4):79:1–79:12, 2011. 7

[19] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006. 2

[20] O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, and D. Cremers. Non-rigid puzzles. In *Computer Graphics Forum*, volume 35, pages 135–143. Wiley Online Library, 2016. 2

[21] S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE TCBB*, 1:24–44, 2004. 1, 2, 3

[22] J. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 1999. 3

[23] F. Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011. 2

[24] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988. 3

[25] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello. Survey of multiobjective evolutionary algorithms for data mining: Part ii. *Evolutionary Computation, IEEE Transactions on*, 18(1):20–35, 2014. 2

[26] M. Nadif and G. Govaert. Model-based co-clustering for continuous data. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, pages 175–180. IEEE, 2010. 2

[27] D. Nogneng and M. Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, 2017. 7

[28] A. Oghabian, S. Kilpinen, S. Hautaniemi, and E. Czeizler. Biclustering methods: Biological relevance and application in gene expression analysis. *PloS one*, 9(3), 2014. 2, 3

[29] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 1, 2, 7

[30] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*. ACM, 2016. 2

[31] S. Pledger and R. Arnold. Multivariate methods using mixtures: Correspondence analysis, scaling and pattern-detection. *Computational Statistics & Data Analysis*, 71:241–261, 2014. 2

[32] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. *Computer Graphics Forum*, 32(2pt4):459–468, 2013. 2

[33] M. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, 1969. 6

[34] R. Priam, M. Nadif, and G. Govaert. Gaussian topographic co-clustering model. In *International Symposium on Intelligent Data Analysis*, pages 345–356. Springer, 2013. 2

[35] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer Graphics Forum*. Wiley Online Library, 2016. 1, 2

[36] E. Savia, K. Puolamäki, and S. Kaski. Latent grouping models for user preference prediction. *Machine Learning*, 74(1):75–109, 2009. 2

[37] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66, 2015. 2

[38] J. Solomon, A. Nguyen, A. Butscher, M. Ben-Chen, and L. Guibas. Soft maps between surfaces. In *Computer Graphics Forum*, volume 31, pages 1617–1626. Wiley Online Library, 2012. 1, 2

[39] J. Solomon, G. Peyré, V. Kim, and S. Sra. Entropic metric alignment for correspondence problems. 2016. 2

[40] Z. Su, Y. Wang, R. Shi, W. Zeng, J. Sun, F. Luo, and X. Gu. Optimal mass transport for shape matching and comparison. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2246–2259, 2015. 2

[41] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. In *ACM transactions on graphics (TOG)*, volume 24, pages 553–560. Acm, 2005. 7

[42] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2013. 1

[43] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011. 1, 2

[44] D. Vu and M. Aitkin. Variational algorithms for biclustering models. *Computational Statistics & Data Analysis*, 89:12–24, 2015. 2