

A Riemannian approach to graph embedding

Antonio Robles-Kelly^{a,*}, Edwin R. Hancock^b

^aNICTA, Building 115, ANU, ACT 0200, Australia¹

^bDepartment of Computer Science, University of York, York YO1 5DD, UK

Received 20 April 2006; accepted 24 May 2006

Abstract

In this paper, we make use of the relationship between the Laplace–Beltrami operator and the graph Laplacian, for the purposes of embedding a graph onto a Riemannian manifold. To embark on this study, we review some of the basics of Riemannian geometry and explain the relationship between the Laplace–Beltrami operator and the graph Laplacian. Using the properties of Jacobi fields, we show how to compute an edge-weight matrix in which the elements reflect the sectional curvatures associated with the geodesic paths on the manifold between nodes. For the particular case of a constant sectional curvature surface, we use the Kruskal coordinates to compute edge weights that are proportional to the geodesic distance between points. We use the resulting edge-weight matrix to embed the nodes of the graph onto a Riemannian manifold. To do this, we develop a method that can be used to perform double centring on the Laplacian matrix computed from the edge-weights. The embedding coordinates are given by the eigenvectors of the centred Laplacian. With the set of embedding coordinates at hand, a number of graph manipulation tasks can be performed. In this paper, we are primarily interested in graph-matching. We recast the graph-matching problem as that of aligning pairs of manifolds subject to a geometric transformation. We show that this transformation is Procrustean in nature. We illustrate the utility of the method on image matching using the COIL database. © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Graph embedding; Riemannian geometry; Combinatorial Laplacian

1. Introduction

The problem of embedding relational structures onto a Riemannian manifold for the purposes of representation or visualisation is an important one in computer science which arises in parameterisation of three-dimensional data [1], multidimensional scaling (MDS) [2] and graph drawing [3]. In general, this problem is clearly one of a combinatorial nature which has been traditionally solved by viewing the edge-weights for the graph as distances between pairs of nodes. The embedding coordinates are then those

corresponding to the isometric mapping of these pairwise distances onto an n -dimensional Euclidean space.

Once the nodes of a graph have been embedded on a manifold, then the graph can be manipulated by applying geometric methods to the embedded points. For instance, by altering the curvature of the manifold, then the graph may be deformed to emphasise different aspects of the set of adjacency relations, and this may be useful for the purposes of visualisation. The problem of locating a spanning tree becomes that of searching for a geodesic path. Finally, the problem of graph-matching can be transformed into that of point-set alignment.

In general, of course, the manifold on which the graph is embedded can have a rather complex structure. In this paper, our aim is to explore whether a simple embedding that restricts the manifold to be of constant sectional curvature can yield useful results. The choice of the sign of the sectional gives considerable freedom, and, as we will see later, the embeddings that result from the choice of negative

* Corresponding author. Tel.: +61 261251384.

E-mail addresses: antonio.robles-kelly@nicta.com.au

(A. Robles-Kelly), erh@cs.york.ac.uk (E.R. Hancock).

¹ National ICT Australia is funded by the Australian Governments Backing Australia's Ability initiative, in part through the Australian Research Council.

curvature and a hyperbolic geometry are very different to those that result from a positive curvature and an elliptic geometry.

Although we also investigate the problems of relational deformation and spanning tree recovery as applications of our embedding method, in this paper we focus mainly on the problem of matching. Hence, in the remainder of this introductory section we review the literature related to embedding graphs onto manifolds, and then focus on how the graph-matching problem can be posed as one of embedding. Finally, we detail the elements of our contribution.

1.1. Related literature

In the mathematics literature, there is a considerable body of work aimed at understanding how graphs can be embedded on a manifold so as to minimise a measure of distortion. Broadly speaking there are three ways in which the problem can be addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph [4]. Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph [5]. A review of methods for efficiently computing distance via embedding is presented in the recent paper of Hjaltason and Samet [6].

In the pattern analysis community, there has recently been renewed interest in the use of embedding methods motivated by graph theory. One of the best known of these is ISOMAP [7]. Here a neighbourhood ball is used to convert data-points into a graph, and Dijkstra's algorithm is used to compute the shortest (geodesic) distances between nodes. By applying MDS to the matrix of geodesic distances the manifold is reconstructed. The resulting algorithm has been demonstrated to locate well-formed manifolds for a number of complex data sets. Related algorithms include locally linear embedding which is a variant of PCA that restricts the complexity of the input data using a nearest neighbour graph [8], and the Laplacian eigenmap that constructs an adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix (the degree matrix minus the weight matrix) [9]. Hein et al. [10] have established the degree of point-wise consistency for graph Laplacians with data-dependent weights to a weighted Laplace operator. Collectively, these methods are sometimes referred to as manifold learning theory.

Embedding methods can also be used to transform the graph-matching problem into one of point-pattern alignment. There is a considerable literature on the problem of graph-matching, and many contrasting methods including search [11], relaxation [12] and optimisation [13] have been successfully used. However, the main challenge in graph-matching is how to deal with differences in node and edge structure. One of the most elegant recent approaches to the graph-matching problem has been to use graph spectral

methods [14], and exploit information conveyed by the eigenvalues and eigenvectors of the adjacency matrix. There have been successful attempts to use spectral methods for both structural graph-matching [15], and for point-pattern matching [16,17]. For instance Umeyama [15] has developed a method for finding the permutation matrix which best matches pairs of weighted graphs of the same size, by using a singular value decomposition of the adjacency matrices. Scott and Longuet-Higgins [16], on the other hand, align point-sets by performing singular value decomposition on a point association weight matrix. Shapiro and Brady [17] have reported a correspondence method which relies on measuring the similarity of the eigenvectors of a Gaussian point-proximity matrix. Although Umeyama's algorithm [15] is elegant in its formulation and can deal with both weighted or unweighted graphs, it cannot be applied to graphs which contain different numbers of nodes and for weighted graphs the method is susceptible to weight errors. One way to overcome these problems is to cast the problem of recovering correspondences in a statistical setting using the EM algorithm [18]. However, the resulting algorithm is time consuming because of its iterative character.

Spectral methods can be viewed as embedding the nodes of a graph in a space spanned by the eigenvectors of the adjacency matrix. In the case of the Umeyama algorithm [15], matching is effected by finding the transformation matrix that best aligns the embedded points. Shapiro and Brady [17] algorithm finds correspondences by seeking the closest embedded points. Kosinov and Caelli [19] have improved this method by allowing for scaling in the eigenspace. The resulting graph-matching methods effectively involve locating correspondences using a simple closest point alignment principal. However, there is clearly scope for using more sophisticated alignment methods. The task of matching point patterns is of pivotal importance in a number of problems in image analysis and pattern recognition. The problem is to find matches between pairs of point-sets when there is noise, geometric distortion and structural corruption. The problem arises in shape analysis [20], motion analysis [21] and stereo reconstruction [17]. Alignment is concerned with the recovery of the geometric transformation that minimises an error functional. If the transformation is a simple isometry, then one of the most elegant and effective methods is to use Procrustes alignment [22,23]. This involves co-centring and scaling the point-sets so that they have the same variance. The rotation matrix is found by performing singular value decomposition on the point correlation matrix, and taking the outer product of the singular vectors.

1.2. Contribution

Our aim in this paper is to seek an embedding of the nodes of a graph which allows matching to be effected using simple point-pattern matching methods. Our observation is that despite proving effective, the traditional spectral

approach to graph embedding is often based upon assumptions concerning the graph topology, such as planarity [24] or orthogonality of the edge-set [25]. Furthermore, it overlooks the relation between the matrix representations of the graph and the differential geometry of the manifold on which the nodes of the graph are embedded. In particular, we aim to draw on the field of mathematics known as spectral geometry, which aims to characterise the properties of operators on Riemannian manifolds using the eigenvalues and eigenvectors of the Laplacian matrix [26]. In doing this, we aim to exploit the link between the graph Laplacian [14] and the Laplace–Beltrami operator [27] for the manifold on which the nodes of the graph reside. The geometric information provided by the spectrum of the Laplace–Beltrami operator is potentially vast. For instance, the first non-zero eigenvalue is important since it provides bounds on both the Ricci and the sectional curvature [27]. The eigenvalues of the Laplace–Beltrami operator can be used to determine both the volume and the Euler characteristic of the manifold. Moreover, the derivative of the zeta function (i.e. the sum of exponential functions with bases equal to the Laplacian eigenvalues) is related to the torsion-tensor for the manifold. Last, but not least, the eigenvalues and eigenfunctions are also of interest since they provide solutions to both the heat and wave equations, and can be used to compute the associated kernel functions.

The aim in this paper is to exploit this relationship between the spectrum of the graph, or combinatorial, Laplacian and the geometry of the corresponding manifold to develop a means of embedding the nodes of a graph on a manifold of constant sectional curvature. The adopted approach is as follows. Using the properties of the Jacobi field associated with the manifold we show how to compute the sectional curvature associated with the geodesic between a pair of points on the manifold. We associate with the geodesic a cost based on length and sectional curvature, and this is used as an edge-weight in our representation. From the edge-weight matrix we compute the combinatorial Laplacian. We then establish the relationship between the Laplacian matrix and the Laplace–Beltrami operator for the manifold making use of the incidence mapping. We do this by viewing the incidence mapping for the graph as the combinatorial analogue of exterior differentiation on manifolds. This treatment, lends itself to the construction of the Laplace–Beltrami operator through the adjoint of the incidence mapping. To perform the embedding of the nodes on the graph onto the manifold, we adopt a method similar to kernel PCA. From the Laplacian, we perform a double centring procedure, and compute Gram matrix of inner products. The eigenvectors of the Gram matrix provide us with the embedding co-ordinates. This embedding procedure is isometric, i.e. it gives us a bijective map that preserves distance.

We exploit this embedding for a number of tasks, however, the most important of these is that of matching. The idea is to embed the nodes of a graph into a metric space and view the graph edge-set as geodesics between pairs of points

in a Riemannian manifold. Viewed in this manner, we can recast the problem of matching the nodes of pairs of graphs as that of aligning their corresponding embedded point-sets. Since our embedding procedure is isometric, a natural way to perform matching is to use Procrustes alignment on the embedded nodes. Further, this alignment process involves only rotation due to centring of the Gram matrix used to recover the embedding coordinates for the nodes in the graph.

This approach has a number of advantages. Firstly, our definition of the edge-weight is linked directly to the geometry of the underlying manifold. Secondly, the relationship between the Laplace–Beltrami operator and the graph Laplacian provides a clear link between Riemannian geometry and spectral-graph theory [14]. Graph-spectral methods have recently proved highly effective in image processing and computer vision [28,29]. By relating the apparatus of graph-spectral theory to Riemannian geometry, making use of the incidence mapping of the graph and the Laplace–Beltrami operator, the results presented here allow a better understanding of these methods. Finally, the recovery of the embedding coordinates and the geometric transformation via linear algebra yields an analytical solution which is devoid of free parameters.

It is important to stress that, although Belkin and Niyogi [9] and Lafferty and Lebanon [30] have proposed methods elsewhere in the literature where the relationship between the graph Laplacian and the Laplace–Beltrami operator has been used for purposes of embedding the graph into a metric space, our method differs from theirs in a number of ways. Firstly, in contrast with the studies in Refs. [9,30], where the linear decomposition of the graph Laplacian is the goal of computation as a means of recovering the embedding of the graph into a metric space, here we go further and both, establish a relationship between the embedding and the intrinsic geometry of the underlying manifold and use the embedding to pose the graph-matching problem as an alignment of manifolds under geometric transformation. Secondly, here we use the double-centred graph Laplacian as an alternative to the non-centred Laplacian for purposes of embedding the graph nodes into a metric space. As a result, the embedding is isometric. This formalism also translates the basis of the embedding space to the origin and allows us to cast the problem of relational matching as an alignment one which involves only rotation. Finally, the work above overlooks the curvature of the manifold in which the nodes are to be embedded, whereas, in the method presented here, the relation between the embedding, the edge-weights and the manifold curvature is made explicitly.

The outline of this paper is as follows. In Section 2, we review the differential geometry of manifolds and explain the geometric ideas underpinning our embedding method. Section 3 details our Procrustes alignment method. In Section 4, we provide illustrations and experiments to demonstrate the utility of our method. Finally, Section 5 offers some conclusions and suggests directions for future work.

2. Riemannian geometry

In this section, we provide the theoretical basis for our graph embedding method. Our aim is to embed the graph nodes as points on a Riemannian manifold. We do this by viewing pairs of adjacent nodes in a graph as points connected by a geodesic on a manifold. In this way, we can make use of Riemannian invariants to compute the cost associated with the edge-set of the graph, i.e. to the embedded point pattern on the manifold. With this characterisation at hand, we show how the properties of the Laplace–Beltrami operator can be used to recover a matrix of embedding coordinates. We do this by establishing a link between the Laplace–Beltrami operator and the graph Laplacian. This treatment allows us to relate the graph Laplacian to a Gram matrix of scalar products, whose entries are in turn related to the squared distances between pairs of points on the Riemannian manifold. In this way, matrix factorisation techniques may then be used to recover the embedding coordinates for the nodes of the graph.

The remainder of this section is divided into three parts. In Section 2.1, we present the necessary elements of Riemannian geometry required to formulate our method. In Section 2.2, we review the definition of the graph Laplacian and its relationship with the Laplace–Beltrami operator. We do this making use of the results on the incidence mapping found in the mathematics literature. In Section 2.3, we show how classical MDS [31,32] can be used to recover the coordinates for the node embedding.

2.1. Riemannian manifolds

In this section, we aim to provide a means of characterising the edges of a graph using a geodesic on a Riemannian manifold. The weight of the edge is the cost or energy associated with the geodesic. To commence, let $G = (V, E, W)$ denote a weighted graph with index-set V , edge-set $E = \{(u, v) | (u, v) \in V \times V, u \neq v\}$ and the edge-weight function set $W: E \rightarrow [0, 1]$. If the nodes in the graph are viewed as points on the manifold, then the weight $W_{u,v}$ associated with the edge connecting the pair of nodes u and v can be interpreted as the energy \mathcal{E}_{p_u, p_v} over the geodesic connecting the pair of points p_u and p_v on the manifold.

To express the energy \mathcal{E}_{p_u, p_v} in terms of geometric invariants, we employ the theory of Jacobi vector fields and their relation to the curvature tensor. In this way, we can characterise the sectional curvature along a geodesic on the manifold. The reasons for using the curvature tensor are twofold. Firstly, the curvature tensor is natural, i.e. it is invariant under isometries (i.e. bijective mappings that preserve distance). Secondly, the curvature tensor measures the anticommutativity of the covariant derivative and is completely determined by the Riemannian metric. Further, the curvature tensor is independent of the choice of coordinate system for the manifold. Hence, the curvature tensor is one of the main invariants in Riemannian geometry.

To commence our development, we review the necessary elements of differential geometry [26,27,33–36]. We consider a function f to be differentiable if it is of class C^∞ , i.e. all its partial derivatives, of all orders, exist and are continuous. Consider a n -dimensional differentiable manifold M . For any point $p \in M$, let M_p denote the tangent space of M at p . Further, let Y be a differentiable vector field in \mathfrak{R}^n such that $Y = \sum_{i=1}^n \eta^i \partial_i$, where η^i is the i th coordinate of the vector $\eta = \sum_{i=1}^n \eta^i \partial_{i|p}$. To provide a definition of ∂_i , we turn our attention to the natural identification $\mathfrak{S}_p: \mathfrak{R}^n \mapsto (\mathfrak{R}^n)_p$ of the tangent space at p , i.e. $(\mathfrak{R}^n)_p$, onto \mathfrak{R}^n . For the natural basis $e = \{e_1, e_2, \dots, e_n\}$ of \mathfrak{R}^n , the chart of the identity map of \mathfrak{R}^n to itself is then given by $\partial_{i|p} = \mathfrak{S}_p e_i$. As a result, the symbol ∂_i is defined so as to be consistent with both, the notion of a chart in Riemannian geometry and the natural basis e of \mathfrak{R}^n .

Throughout the paper, we will require a well-defined method for differentiating vector fields. Hence, for a collection of vector fields \wp^1 of class C^1 and a differentiable vector $\xi \in M_p$, the connection $\nabla: M_p \times \wp^1(M_p) \mapsto M_p$ is given by $\nabla_\xi Y = \sum_{i=1}^n (\xi \eta^i) \partial_i$. This definition implies that the vector $\nabla_\xi Y$ is in the same tangent space as ξ . Furthermore, the connection expresses the covariant derivatives of the vector field Y in terms of the vector ξ . That is, $\nabla_\xi Y$ describes the rate of change of the vector field Y in the direction ξ in terms of ξ itself.

Let the vector fields Y, X and Z be the extensions over a neighbourhood of p of the vectors $\eta, \xi, \zeta \in M_p$. The curvature tensor, which is quadrilinear in nature [26], is denoted by $R(\xi, \eta)\zeta$. Here, we are interested in the sectional curvature, which is bilinear in nature. To obtain a bilinear form, i.e. the sectional curvature, from the curvature tensor we use two linearly independent vectors $\eta, \xi \in M_p$ and write

$$\mathcal{K}(\xi, \eta) = \frac{\langle R(\xi, \eta)\xi, \eta \rangle}{|\xi|^2 |\eta|^2 - \langle \xi, \eta \rangle^2}. \quad (1)$$

As mentioned earlier, we are interested in modelling the edges in the graph as geodesics on a manifold. Consider the parameterised curve $\gamma: t \in [\alpha, \beta] \mapsto M$. From Riemannian geometry, we know that for γ to be a geodesic, it must satisfy the condition $\nabla_{\gamma'} \gamma' = 0$. It can be shown that the connection ∇ for geodesics is, in fact, a Levi–Civita connection [26]. Further, Levi–Civita connections are metric preserving, unique and are guaranteed to exist.

To take our analysis further, we define the Jacobi field along γ as the differentiable vector field $Y \in M_p$, orthogonal to γ , satisfying Jacobi’s equation

$$\nabla_t^2 + R(\gamma', Y)\gamma' = 0, \quad (2)$$

where ∇ is a Levi–Civita connection.

With these ingredients, we can substitute ξ and η with γ' and Y in Eq. (1) and write

$$\mathcal{K}(\gamma', Y) = \frac{\langle R(\gamma', Y)\gamma', Y \rangle}{|\gamma'|^2 |Y|^2 - \langle \gamma', Y \rangle^2}. \quad (3)$$

Because Y is orthogonal to γ' , the equation above becomes

$$\mathcal{K}(\gamma', Y) = \frac{\langle R(\gamma', Y)\gamma', Y \rangle}{|\gamma'|^2|Y|^2}. \tag{4}$$

To simplify the expression for the sectional curvature further, we make use of the fact that, since Y is a Jacobi field, it must satisfy the condition $\nabla_t^2 Y = -R(\gamma', Y)\gamma'$. Hence, we can write

$$\mathcal{K}(\gamma', Y) = \frac{\langle -\nabla_t^2 Y, Y \rangle}{\langle Y, Y \rangle}, \tag{5}$$

where we have substituted $|Y|^2$ with $\langle Y, Y \rangle$ and set $|\gamma'| = 1$. As a result, it follows that $\nabla_t^2 Y = -\mathcal{K}(\gamma', Y)Y$. Hence, the Laplacian operator $\nabla_t^2 Y$ is determined by the sectional curvature of the manifold.

This suggests a way of formulating the energy over the geodesic $\gamma \in M$ connecting the pair of points corresponding to the nodes indexed u and v . Consider the geodesic γ subject to the Jacobi field Y . The energy over the geodesic γ can be expressed making use of the equations above as

$$\begin{aligned} \mathcal{E}(p_u, p_v) &= \int_{\gamma} |\gamma' + \nabla_t^2 Y|^2 dt \\ &= \int_{\gamma} |\gamma' - \mathcal{K}(\gamma', Y)Y|^2 dt. \end{aligned} \tag{6}$$

We can provide a physical interpretation of the above result. Consider the term given by $\int_{\gamma} |\gamma'|^2 dt$, which is the square of the arc-length for the geodesic γ . This can be viewed as the potential energy along the geodesic from the point indexed u to the point indexed v . Now, we turn our attention to the term $\int_{\gamma} |\nabla_t^2 Y|^2 dt$, which can be regarded as the kinetic energy excerpted by the Jacobi field over γ . The quantity in Eq. (6) is then the sum of both, the potential and kinetic energies for the geodesic from the point indexed u to the point indexed v . As a result, the edge-weight is small if a pair of points are close to one another or the curvature along the geodesic between them is small.

In practice, we will confine our attention to the problem of embedding the nodes on a constant sectional curvature surface. For such a surface, the sectional curvature is constant i.e. $\mathcal{K}(\gamma', Y) \equiv \kappa$. Under this restriction, the Jacobi field equation (see Eq. (2)) becomes

$$\nabla_t^2 Y = -\kappa Y. \tag{7}$$

With the boundary conditions $Y(0) = 0$ and $|\nabla_t Y(0)| = 1$, the solution is

$$Y(t) = \begin{cases} \frac{\sin(\sqrt{\kappa}t)}{\sqrt{\kappa}}\eta & \text{if } \kappa > 0, \\ t\eta & \text{if } \kappa = 0, \\ -\frac{\sinh(\sqrt{-\kappa}t)}{\sqrt{-\kappa}}\eta & \text{if } \kappa < 0, \end{cases} \tag{8}$$

where the vector η is in the tangent space of M at p_u and is orthogonal to γ' at the point indexed u , i.e. $\eta \in M_{p_u}$ and $\langle \eta, \gamma'|_{p_u} \rangle = 0$.

With these ingredients, and by rescaling the parameter t so that $|\gamma'| = a$, we can express the weight of the edge connecting the nodes indexed u and v as follows:

$$W(u, v) = \begin{cases} \int_0^1 (a(u, v)^2 + \kappa(\sin(\sqrt{\kappa}a(u, v)t))^2) dt & \text{if } \kappa > 0, \\ \int_0^1 a(u, v)^2 dt & \text{if } \kappa = 0, \\ \int_0^1 (a(u, v)^2 - \kappa(\sinh(\sqrt{-\kappa}a(u, v)t))^2) dt & \text{if } \kappa < 0, \end{cases} \tag{9}$$

where $a(u, v)$ is the Euclidean distance between each pair of points in the manifold, i.e. $a(u, v) = \|p_u - p_v\|$.

2.2. The graph Laplacian

In this section, we demonstrate the relationship between the graph Laplacian and the Laplace–Beltrami operator. It is worth noting that, in this paper, we follow von Luxburg et al. [37] and use the normalised Laplacian as an alternative to the non-normalised one. The relationship between the discrete Laplacian of a graph, and the continuous Laplace–Beltrami operator has been extensively studied in the literature. It has been well known for some time that as the number of sample points or nodes increases, then the discrete operator converges towards the continuous one [38]. Hein et al. [10] present a more recent analysis of convergence which draws on the concept of pointwise consistency. Their analysis considers both the variance and bias of the estimate of continuous Laplace–Beltrami operator provided by the discrete Laplacian when particular choices of the adjacency weight matrix are made. Belkin and Niyogi [39] consider the conditions under which the graph Laplacian of a point cloud approaches the Laplace–Beltrami operator. If there are n sample points and the Laplacian is computed on a k -dimensional manifold, then the rate of convergence is $O(n^{-k+2/(2(k+2+\alpha))}) \simeq O(n^{-1/2})$ where $\alpha > 0$. Interestingly, although the analysis relies on the concept of manifold curvature, the final convergence behaviour is independent of the manifold curvature. Finally, Singer [40] improves the analysis of bias and variance provided by Hein et al. [10] and determines the dependence of the convergence rate on the kernel bandwidth ε used in the adjacency weighting function. The bias term is shown to depend on the curvature of the manifold. The analysis of the variance term shows that the convergence rate is $O(n^{-1/2}\varepsilon^{-\kappa/4})$.

To commence, recall that the weight matrix W is related to the normalised Laplacian $\mathcal{L} = \mathbf{D}^{-1/2}(\mathbf{D} - W)\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, where \mathbf{D} is a diagonal matrix such that $\mathbf{D} = \text{diag}(\text{deg}(1), \text{deg}(2), \dots, \text{deg}(|V|))$ and $\text{deg}(v)$ is the degree of the node indexed v , i.e. $\text{deg}(v) = \sum_{v \sim u; u \in V} W(v, u)$. Further, as noted by Chung [14], we can view the graph Laplacian \mathcal{L} as an operator over the set of real-valued functions $f: V \mapsto \mathfrak{R}$ such that, for a pair of nodes, u and $v \in V$

corresponding to the points p_u and $p_v \in M$, we have

$$\mathcal{L}f(v) = \sum_{v \sim u} \frac{W(v, u)}{\sqrt{\deg(u)\deg(v)}} (f(v) - f(u)). \quad (10)$$

It is worth noting in passing that, in general, the term associated to the curvature $\mathcal{K}(\gamma', Y)Y$ in Eq. (6) is asymmetric. This follows from the bilinear form of the sectional curvature and the directional nature of the Jacobi field Y . As a result, the graph whose edge-weights are given by $W(u, v) = \mathcal{E}(p_u, p_v)$ is, potentially, a directed one. However, our analysis concerns undirected graphs whose edge-weights are real and non-negative. This is due to the fact that, as a result of embedding the nodes into a manifold of constant sectional curvature, the matrix W is always symmetric and, therefore, the matrix $\mathcal{L} = \mathbf{D}^{1/2}(\mathbf{D} - W)\mathbf{D}^{-1/2}$ is guaranteed to be positive, semidefinite.

We define a mapping \mathcal{I} of all functions on the set of vertices V to the functions $g(e)$ over the set of edges E . The incidence mapping \mathcal{I} is then an operator such that

$$\mathcal{I}g(e) = f(e_+) - f(e_-), \quad (11)$$

where the nodes $v = e_+$ and $u = e_-$ are the head and tail, respectively, of the edge $e \in E$ and $g: E \mapsto \mathfrak{R}$ is a real-valued function. Thus, \mathcal{I} maps functions in the set of edges E to functions in the set of vertices V . Further, the incidence mapping \mathcal{I} is a $|V| \times |E|$ matrix which satisfies

$$\mathcal{L} = \mathcal{I}\mathcal{I}^T. \quad (12)$$

From the expression above, we can make use of the adjoint \mathcal{I}^* of the incidence mapping \mathcal{I} and write

$$\mathcal{I}^*f(v) = \sum_{e \in E; e_+ = v} g(e). \quad (13)$$

In the equation above, we have written $e_+ = v$ to imply that the head of the edge incident in v is on the tangent space to M at p_v . Further, the adjoint \mathcal{I}^* of the incidence mapping \mathcal{I} can be viewed as an operator which generalises the conjugate transpose of a square matrix [41].

Since the Laplace–Beltrami operator is a self-adjoint one, we can establish a relationship between exterior differentiation d on Riemannian manifolds and the incidence mapping \mathcal{I} through the expression $\Delta = \delta d$, where $\delta = - * d *$ and $*$ is the Hodge star operator, i.e. a linear map between a pair of vector spaces in k and $k - n$ dimensions [42]. The Hodge star operator is then a correspondence mapping which can be regarded as a linear coordinate transformation of differential forms [43]. Thus, we can view the adjoint \mathcal{I}^* of the incidence mapping as the combinatorial analogue of the exterior differentiation on manifolds and express the combinatorial Laplacian as follows:

$$\mathcal{L} = \mathcal{I}\mathcal{I}^* \quad (14)$$

which is equivalent to Eq. (12) for real, symmetric incidence mappings.

In fact, Zucker [38] conjectured that if M is a complete Riemannian manifold of finite volume, such that the curvature tensor $R(\zeta, \eta)\zeta$ is bounded on M , then the Laplace–Beltrami operator Δ has a closed range on the L^2 space of square summable p -forms of d^* , i.e. the rank- p differential forms of the exterior differentiation adjoint. This is consistent with the developments in Refs. [10,40,39]. Furthermore, as shown by Zucker [38], it guarantees the existence of the parametrix for the Laplace–Beltrami operator. The parametrix is an expansion of functions whose asymptotic behaviour is the same as that of Δ [42]. Thus, one of the main immediate consequences of this conjecture is to provide a guarantee for pointwise consistency between Δ and \mathcal{L} for large sample point-sets, i.e. $O(n^{-1/2}) \rightarrow 0$ as $n \rightarrow \infty$.

2.3. Recovery of the embedding coordinates

To construct a set of embedding coordinates for the nodes of the graph, we use classical MDS with double centring [31]. We do this so as to cast the recovery of the embedding coordinates for the nodes in the graph in terms of a matrix transformation of the graph Laplacian \mathcal{L} into a known canonical form. Since the double centring procedure introduces a linear dependency over the columns of the matrix [2], the double-centred graph Laplacian \mathbf{H} is, in fact, a Gram matrix and, thus, we can recover the node-coordinates making use of a matrix decomposition approach. The double centring operation on the graph Laplacian also has the effect of translating the coordinate system for the embedding to the origin.

The double-centred graph Laplacian is then given by

$$\mathbf{H} = -\frac{1}{2}\mathbf{B}\mathcal{L}\mathbf{B}, \quad (15)$$

where

$$\mathbf{B} = \mathbf{I} - \frac{1}{|V|}\mathbf{e}\mathbf{e}^T$$

is the centring matrix, \mathbf{I} is the identity matrix and \mathbf{e} is the all-ones vector, i.e. a vector whose elements are all identical to unity. The element indexed v, w of the matrix \mathbf{H} is then given by

$$\mathbf{H}(u, v) = -\frac{1}{2}[\mathcal{L}(u, v)^2 - \mathcal{A}^2 - \mathcal{B}^2 + \mathcal{C}^2], \quad (16)$$

where

$$\mathcal{A} = \frac{1}{|\Omega_v|} \sum_{u \sim v; w \in V} \mathcal{L}(u, w),$$

$$\mathcal{B} = \frac{1}{|\Omega_u|} \sum_{v \sim u; w \in V} \mathcal{L}(w, v),$$

$$\mathcal{C} = \frac{1}{|V|^2} \sum_{u, v \in V} \mathcal{L}(u, v), \quad (17)$$

and Ω_v is the set of first-neighbours of the node $v \in V$. From the above equations, $\mathcal{A}f(u)$ can be regarded as the average value of the functions $\mathcal{I}f(e) = f(e_+) - f(e_-)$ over those nodes w adjacent to the node u . Similarly, $\mathcal{B}f(v)$ is the average value of the functions $-\mathcal{I}f(e) = f(e_-) - f(e_+)$ over those nodes $w \sim v$. The average value over the set of functions $\mathcal{I}f$ is given by $\mathcal{C}f$.

The matrix of embedding coordinates \mathbf{J} is obtained from the centred Laplacian using the factorisation $\mathbf{H} = \mathbf{J}\mathbf{J}^T$. To perform this factorisation we make use of Young–Householder theorem [32]. Let $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{|V|})$ be the diagonal matrix with the ordered eigenvalues of \mathbf{H} as elements and $\Phi = (\phi_1 | \phi_2 | \dots | \phi_{|V|})$ be the matrix with the corresponding ordered eigenvectors as columns. Here the ordered eigenvalues and corresponding eigenvectors of the matrix \mathbf{H} satisfy the condition $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_{|V|}| > 0$. As a result we can write $\mathbf{H} = \Phi \Lambda \Phi^T = \mathbf{J}\mathbf{J}^T$, where $\mathbf{J} = \sqrt{\Lambda} \Phi$. The matrix which has the embedding co-ordinates of the nodes as columns is $\mathbf{D} = \mathbf{J}^T$. Hence, $\mathbf{H} = \mathbf{J}\mathbf{J}^T = \mathbf{D}^T \mathbf{D}$ is a Gram matrix, i.e. its elements are scalar products of the embedding co-ordinates. Consequently, the embedding of the points is an isometry.

3. Graph matching by point-set alignment

Later on, we will explore the application of the embedding method to a number of problems. However, one of the most important of these is to show how the nodes of the embedded graph can be matched by performing point-set alignment. Hence, in this section, we show how the graph-matching process can be posed as one of manifold alignment. This can be effected by finding the geometric transformation which minimises a quadratic error measure, i.e. least squares distance, between pairs of embedded points. To commence, we require some formalism. Suppose that $\mathbf{H}_D = \Phi_D \Lambda_D \Phi_D^T$ is the centred Laplacian matrix for the set of $|V^D|$ “data” points whose embedded co-ordinates are given by the matrix $\mathbf{D} = \sqrt{\Lambda_D} \Phi_D^T$. Similarly, $\mathbf{H}_M = \Phi_M \Lambda_M \Phi_M^T$ is the centred Laplacian matrix for the set $|V^M|$ of “model” points whose embedded co-ordinates are given by the matrix $\mathbf{M} = \sqrt{\Lambda_M} \Phi_M^T$. In practice, the sets of points to be matched may not be of the same size. To accommodate this feature of the data, we assume that the model point-set is the larger of the two, i.e. $|V^M| \geq |V^D|$. As a result of the Young–Householder factorisation theorem used in the previous section, the embeddings of the data and model point patterns onto the manifolds $M^D \in \mathfrak{R}^{|V^D|}$ and $M^M \in \mathfrak{R}^{|V^M|}$, respectively, will be assumed to have a dimensionality which is equal to the number of points in the corresponding point-set. Hence, in order to be consistent with our geometric characterisation of the point-pattern matching problem, we consider the manifold $M^D \in \mathfrak{R}^{|V^D|}$ to be a covering map, or projection, of the manifold $M^M \in \mathfrak{R}^{|V^M|}$. Here, in order to avoid ambiguities, we are interested in coverings of multiplicity one and, therefore, as an alternative to the matrix

\mathbf{D} , we work with the matrix of coordinates

$$\tilde{\mathbf{D}} = [\mathcal{D} | \mathbf{n}_{|V^D|+1} | \mathbf{n}_{|V^D|+2} | \dots | \mathbf{n}_{|V^M|}], \quad (18)$$

where \mathbf{n}_i is a vector of length $|V^D|$ whose entries are null.

With these ingredients, the problem of finding a transformation which can be used to map the data-points onto their counterparts in the model point-set can be viewed as that of finding the rotation matrix \mathbf{R} and the point-correspondence matrix $\tilde{\mathbf{P}} = [\mathbf{P} | \mathbf{O}]$, where \mathbf{P} is a permutation matrix of order $|V^D|$ and \mathbf{O} is a null matrix of size $|V^D| \times |V^M - V^D|$, which minimise the quadratic error function

$$\varepsilon = \|\mathbf{M} - \tilde{\mathbf{P}}\tilde{\mathbf{D}}\|^2. \quad (19)$$

To solve the problem, we divide it into two parts. First, we find the rotation matrix \mathbf{R} by assuming the point-correspondence matrix $\tilde{\mathbf{P}}$ is known. Second, with the optimum rotation matrix at hand, we recover the point-correspondence matrix $\tilde{\mathbf{P}}$.

To recover the rotation matrix \mathbf{R} , we make use of the fact that both the matrices \mathbf{R} and $\tilde{\mathbf{P}}$ are orthogonal and write

$$\begin{aligned} \varepsilon &= \text{Tr}[\mathbf{M}\mathbf{M}^T] + \text{Tr}[(\tilde{\mathbf{P}}\tilde{\mathbf{D}})(\tilde{\mathbf{P}}\tilde{\mathbf{D}})^T] \\ &\quad - 2 \text{Tr}[\mathbf{M}(\mathbf{R}\tilde{\mathbf{D}})^T \tilde{\mathbf{P}}]. \end{aligned} \quad (20)$$

From the equation above, it is clear that maximising $\text{Tr}[\mathbf{M}(\mathbf{R}\tilde{\mathbf{D}})^T \tilde{\mathbf{P}}]$ is equivalent to minimising ε . Further, assuming that the optimum correspondence matrix $\tilde{\mathbf{P}}$ is known, we can view the matrix $\tilde{\mathbf{P}}$ as an augmented permutation matrix, and, hence maximising $\text{Tr}[\mathbf{M}\mathbf{D}^T \tilde{\mathbf{R}}]$ is the same as maximising $\text{Tr}[\mathbf{M}(\mathbf{R}\tilde{\mathbf{D}})^T \tilde{\mathbf{P}}]$. This observation is important, because it implies that the rotation matrix \mathbf{R} is the solution to a Procrustean transformation over the embedding coordinates for the set of data-points. Recall that a Procrustes transformation is of the form $\mathbf{Q} = \mathbf{R}\tilde{\mathbf{D}}$ which minimises $\|\mathbf{M} - \mathbf{Q}\|^2$. It is known that minimising $\|\mathbf{M} - \mathbf{Q}\|^2$ is equivalent to maximising $\text{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T \mathbf{R}]$. This is effected by using Kristof’s inequality, which states that, if \mathbf{S} is a diagonal matrix with non-negative entries and \mathbf{T} is orthogonal, we have

$$\text{Tr}[\mathbf{T}\mathbf{S}] \geq \text{Tr}[\mathbf{S}]. \quad (21)$$

Let the singular value decomposition (SVD) of $\tilde{\mathbf{D}}\mathbf{M}^T$ be $\mathbf{U}\mathbf{S}\mathbf{V}^T$. Using the invariance of the trace function over cyclic permutation, and drawing on Kristof’s inequality, we can write

$$\text{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T \mathbf{R}] = \text{Tr}[\mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{R}] = \text{Tr}[\mathbf{V}^T \mathbf{R} \mathbf{U}\mathbf{S}] \geq \text{Tr}[\mathbf{S}]. \quad (22)$$

It can be shown that $\mathbf{V}^T \mathbf{R} \mathbf{U}$ is orthogonal since \mathbf{R} is orthogonal. Furthermore, the maximum of $\text{Tr}[\mathbf{M}(\mathbf{R}\tilde{\mathbf{D}})^T \tilde{\mathbf{P}}]$ is achieved when $\mathbf{V}^T \mathbf{R} \mathbf{U} = \mathbf{I}$. As a result, the optimal rotation matrix \mathbf{R} is given by

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T. \quad (23)$$

With the rotation matrix at hand, the correspondence matrix $\tilde{\mathbf{P}}$ can be recovered by noting that the product $\mathbf{M}(\mathbf{R}\tilde{\mathbf{D}})^T = \mathbf{M}\mathbf{Q}^T$ is the matrix of pairwise inner products between

the embedding coordinates for the data and model point-sets. Since the rotation of $\tilde{\mathbf{D}}$ over \mathbf{R} is optimum, the normalised inner product between pairs of matching points is, in the ideal case, equal to unity, i.e. the angle between normalised coordinate vectors is zero. To take advantage of this, we construct the matrix of normalised pairwise inner products and then use it to recover $\tilde{\mathbf{P}}$. Hence, consider the matrix \mathbf{Z} of order $|V^{\mathcal{D}}| \times |V^{\mathcal{M}}|$ whose element indexed i, j is given by the normalised inner product of the respective embedding coordinate vectors, after being aligned by rotation, for the data-point indexed i and j th model-point. The elements of the matrix \mathbf{Z} are hence given by

$$\mathbf{Z}(i, j) = \frac{\sum_{k=1}^{|V^{\mathcal{M}}|} \mathbf{Q}(i, k)\mathbf{M}(j, k)}{\sqrt{\sum_{k=1}^{|V^{\mathcal{M}}|} \mathbf{Q}(i, k)^2} \sqrt{\sum_{k=1}^{|V^{\mathcal{M}}|} \mathbf{M}(j, k)^2}}. \quad (24)$$

Since the correspondence matrix $\tilde{\mathbf{P}}$ can be viewed as a matrix which slots over the matrix \mathbf{Z} of normalised pairwise inner products and selects its largest values, we can recover $\tilde{\mathbf{P}}$ from \mathbf{Z} in the following way. We commence by clearing $\tilde{\mathbf{P}}$ and, recursively, do

1. $\tilde{\mathbf{P}}(i, j) = 1$, where $\{i, j | \mathbf{Z}(i, j) = \max_{z(i, j) \neq 0}(\mathbf{Z})\}$.
2. $\mathbf{Z}(i, k) = 0 \forall k \in |V^{\mathcal{M}}|$ and $\mathbf{Z}(l, j) = 0 \forall l \in |V^{\mathcal{D}}|$.

until $\mathbf{Z} \equiv 0$. The data-point indexed i is then a match to the j th model-point *if and only if* $\tilde{\mathbf{P}}(i, j) = 1$. It is important to note that \mathbf{Z} is the equivalent to the correlation, in a scalar-product sense, between the rows of \mathbf{M} and the columns of \mathbf{Q}^T . It can be shown that the matrix $\tilde{\mathbf{P}}$ maximises the trace of $\tilde{\mathbf{P}}^T \mathbf{M} \mathbf{Q}^T$ and, hence, minimises the quadratic error function ε .

This geometric treatment of the node-correspondence problem and its relationship to the correlation, as captured by the entries of \mathbf{Z} , between the rows of \mathbf{M} and the columns of \mathbf{Q}^T lends itself naturally to further refinement via statistical approaches such as EM algorithm [23] or relaxation labelling [44]. For instance, in the case of the EM algorithm, the required correspondences are the missing or hidden data and the geometric transformation parameters are those that need to be recovered. In Section 4.2, we elaborate further on the use of the EM algorithm for iterative Procrustes alignment and compare the method in Ref. [23] with the one presented above.

4. Experiments

The experimental evaluation of our method is divided into two parts. We commence with an illustration on the effect of the embedding on point-sets. In Section 4.1.1 we show the effect of the embedding procedure on different arrangements of points. In Section 4.1.2, we show how the embedding can be used to extract spanning trees from spatial data. Second, in Section 4.2 we illustrate the effectiveness of the

embeddings for Procrustean alignment. Here we experiment with real-world data provided by the COIL data-base.

4.1. Illustrations

4.1.1. Point-set deformation

In this section, we illustrate the utility of our method for the purposes of embedding a set of data-points in a Riemannian manifold of constant sectional curvature. For this purpose, we have used two sets of points drawn from a two dimensional Euclidean space. The first of these, corresponds to 25 points sampled regularly from a two-dimensional lattice. Our second data set comprises 30 points whose coordinates are binormally distributed on a Euclidean plane. In Fig. 1, we show the sets of points used in our experiments.

In Fig. 2, we show the results obtained by our algorithm. In our experiments, we have set κ to $-10, 0$ and 10 . In the first row, from left-to-right, we show the embedding results, for increasing values of κ , corresponding to the point-lattice in the left-hand panel of Fig. 1. The second row shows the matrices of edge-weights for the embeddings shown in the first row. In the third and fourth rows, we repeat this sequence for the set of points shown in the right-hand panel of Fig. 1.

From the results shown in Fig. 2, it is clear that the sectional curvature κ has an important effect in the recovery of the embedding coordinates. We consider first the effect on the regular point lattice. For $\kappa = 0$ the embedding is just a rotated version of the original distribution of the points in the plane. When κ is non-zero, then different patterns of behaviour emerge. In the case of negative sectional curvature (i.e. hyperbolic geometry), the embedding “collapses” the distribution of points towards the origin. For positive sectional curvature (i.e. elliptic geometry) the effect is to push the points away from the origin, and the point distribution forms an annulus.

A similar behaviour can be seen in the binormally distributed point-set in the right-hand panel of Fig. 1. Here, for $\kappa = -10$, there is again a tendency to move the points towards the origin. For $\kappa = 10$, the embedding enforces a clear separation between the two point-clouds in the data set. This behaviour is consistent with the fact that, for hyperbolic surfaces ($\kappa < 0$) parallel lines diverge. For spherical manifolds ($\kappa > 0$), parallel lines intersect.

Finally, we illustrate further the effect of the curvature in the edge-weights $W(u, v)$. To do this, we have plotted, in Fig. 3, the values of the edge-weights $W(u, v)$, as given in Eq. (9), between a pair of nodes as a function of the Euclidean distance $a(u, v)$ between two nodes in the manifold for $\kappa = -15, -7.5, 0, 7.5, 15$. In the plot, for the sake of clarity, we have scaled the axes logarithmically. From the plot, we can see that, for negative values of curvature, the edge-weight increases exponentially with respect to the distance. For positive sectional curvature, the increase is slower and the edge-weight has a stationary point when $a(u, v) = 1$.

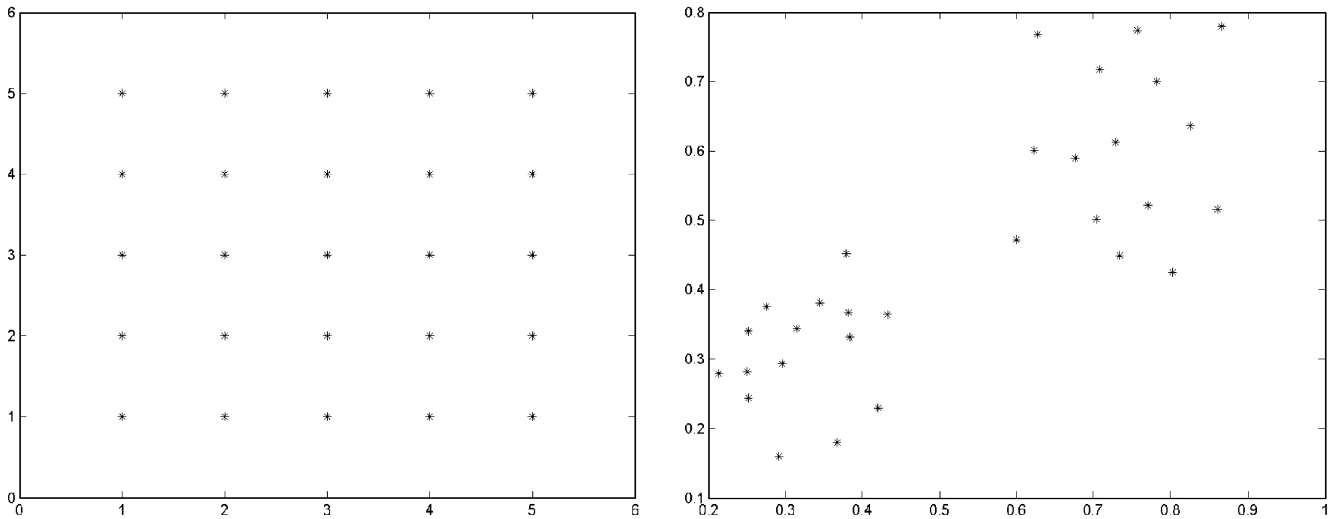


Fig. 1. Sets of points used in our experiments.

As a result, for negative values of sectional curvature, the edge-weights $W(u, v)$ remain small for values of $a(u, v)$ close to zero, but for $\kappa \ll 0$ and $a(u, v)$ close to one, the edge-weight is large.

4.1.2. Minimum spanning tree recovery

In this section, we illustrate the utility of our method for the purposes of network optimisation. Our experimental vehicle is a distribution network in the United Kingdom (UK). For this purpose, we have used a set of 30 points drawn from city locations in the UK. We do this by making use of the postcodes for 30 cities to locate points, by longitude and latitude, on a map of the UK. Our graph is then given by the Delaunay triangulation of these points.

For the embedding of this data, we have computed the minimum spanning tree. To do this, we have searched for the spanning tree of minimum total edge-weight using Dijkstra's algorithm. In Fig. 4, we show the results obtained. In the left-hand panel, we show the Delaunay graph for the 30 cities used in our experiments. The middle panel shows the edge-weight matrix, i.e. the matrix W for the Delaunay graph in the left-hand panel. Finally, the right-hand panel shows the spanning tree recovered by our algorithm. Here, we have indicated the root of the tree with a circle, while the rest of the points have been plotted using asterisks.

4.2. Feature point matching

In this section, our goals are twofold. Firstly, we aim at assessing the quality of the matching results delivered by our algorithm. Secondly, we provide a more comprehensive illustration of the utility of the algorithm for the purposes of similarity-based image retrieval. As an experimental vehicle, we use the Columbia University COIL-20 database. The COIL-20 database contains 72 views for 20 objects acquired

by rotating the object under study about a vertical axis. In Fig. 5, we show sample views for each of the objects in the database. For each of the views, our point patterns are comprised of feature points detected using the Harris corner detector [45].

4.2.1. Matching

To evaluate the results of matching pairs of views in the database, we have adopted the following procedure. For each object, we have used the first 15 views, 3 of these are "model" views and the remaining 12 are "data" views. We have then matched, by setting κ to -15 , 0 and 15 , the feature points for the selected "model" views with those corresponding to the two previous and two subsequent views in the database, i.e. we have matched the feature points for the i th view with those corresponding to the views indexed $i - 2$, $i - 1$, $i + 1$ and $i + 2$.

We have also implemented the iterative Procrustes alignment method presented in Ref. [23]. This is a matching algorithm that poses the correspondence problem as an alignment one whose utility measure is the expected log-likelihood function. Under the assumption that the positional errors between the aligned point-sets are Gaussian, the problem of minimising a weighted Procrustes distance over the embedding coordinates is cast into a maximum likelihood setting. The a posteriori probabilities of the point correspondences are then used to control the different positional errors so as to iterate on a weighted correspondence matrix between two interleaved computational steps. In the expectation step, the a posteriori correspondence probabilities are estimated from the current position errors. This is done by applying the Bayes formula to the Gaussian distribution functions. In the maximisation step, the alignment parameters are estimated so as to maximise the expected log-likelihood function. These two steps are interleaved until convergence is reached.

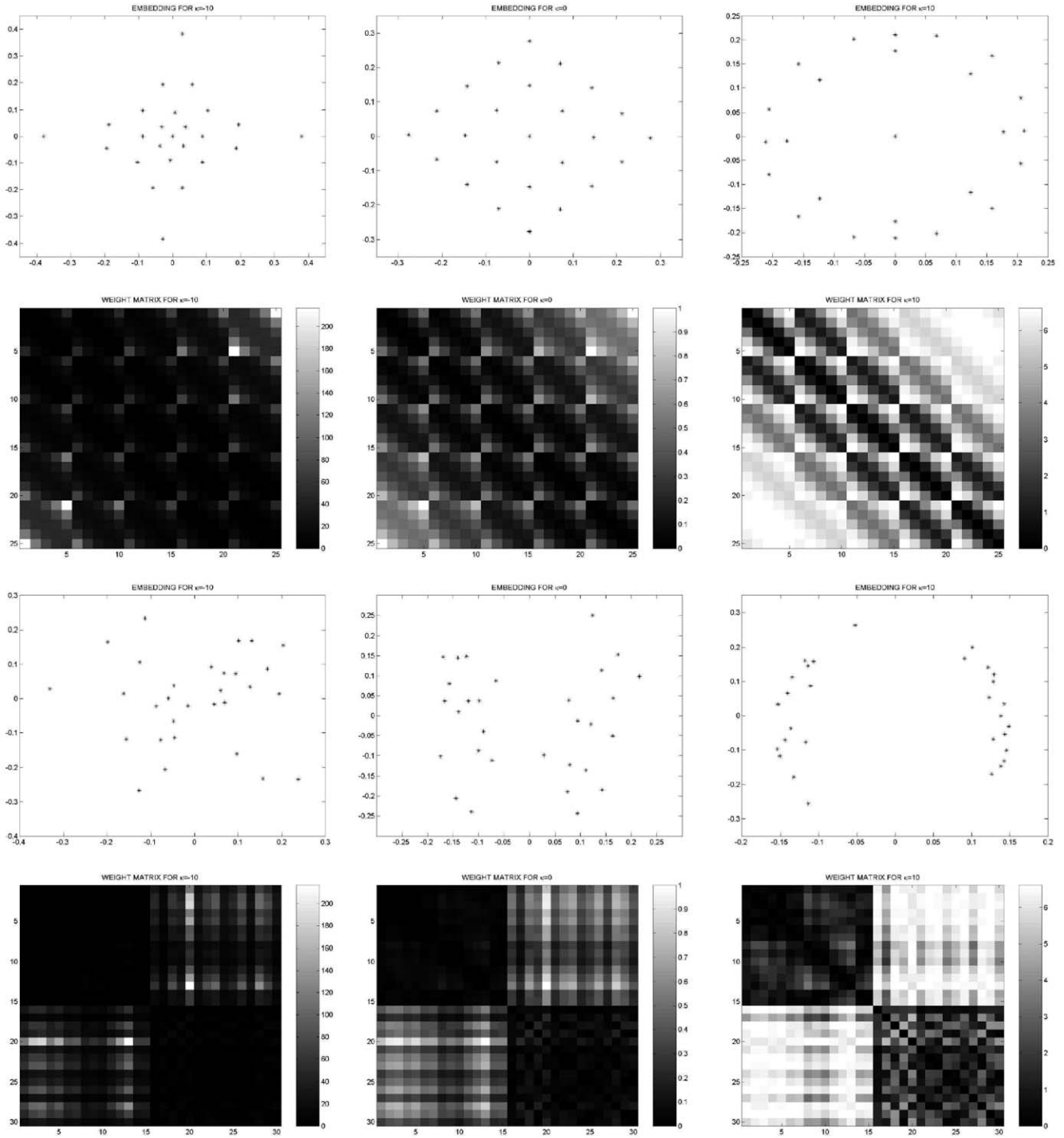


Fig. 2. First row: embeddings for the point-lattice with $\kappa = -10, 0, 10$; second row: matrices of edge-weights for the embeddings in the first row; third row: embedding results with $\kappa = -10, 0, 10$ for the point-set in the right-hand panel of Fig. 1; fourth row: matrices of edge-weights for the embeddings in the third row.

In Fig. 4.2, we show sample matching results for three objects obtained with $\kappa = -15$. In the left-hand column, we show the results for the non-iterative version of the algorithm, i.e. the algorithm as presented in Section 3. The right-hand column shows the results obtained using the iter-

ative Procrustes alignment method in Ref. [23]. In the plots, for the sake of clarity, we only show 20 randomly sampled correspondences from the total number of corners. Here, the squared markers denote a corner while the correspondences are the white lines between the two views in each panel. In

each row, the left-hand-side of the panel shows the “model” view, which is the one indexed 3 for the three objects. The right-hand-side of the panel shows the “data” view, i.e. the second view in the database for each of the three objects. To provide more qualitative results, we have ground-truthed the correspondences between the “model” and “data” views and computed the normalised average ratio ε of incorrect to

correct correspondences $\mu(k, j)$ between the “model” view indexed k and the corresponding “data” view indexed j . The quantity ε is then given by

$$\varepsilon = \frac{1}{4|\Pi|} \sum_{k \in \Pi} \sum_{\substack{j=k+2 \\ j \neq k}}^{j=k+2} \frac{\mu(k, j)}{\rho(k, j)}, \quad (25)$$

where $\Pi = \{3, 6, 9, 12\}$ is the set of indices for the “model” views and $\rho(k, j)$ is the maximum number of correspondences between the “model” and the “data” view.

In Table 1, we show results for both, the non-iterative and the iterative Procrustes alignment algorithms. Note that, in the table, we have used the object indexes in Fig. 5. From the table, we conclude that the value of the sectional curvature κ has an important effect in the results delivered by both methods. These perform consistently better for negative values of κ . This is the case in which the alignment is performed between manifolds that are hyperbolic in nature. We also note that the iterative Procrustes alignment method gives a margin of advantage over the non-iterative algorithm presented in Section 3. It is worth stressing, however, that this is not without a sacrifice in computational complexity and coding difficulty. The iterative alignment method based upon the EM algorithm is more complicated to code and, due to its iterative nature, more computationally burdensome. In our experiments, the EM algorithm took, on a 2.0GHz dual-Xeon workstation, 7.8 times more time to recover the

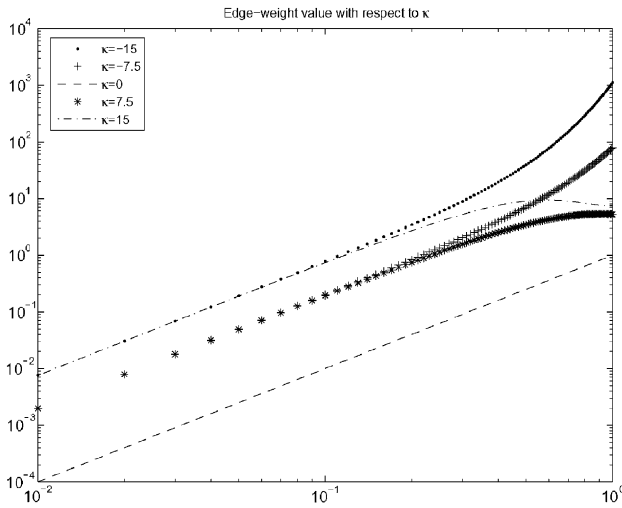


Fig. 3. Value of the edge-weight $W(u, v)$ as a function of the Euclidean distance $a(u, v)$ between a pair of nodes in the manifold for $\kappa = -15, -7.5, 0, 7.5, 15$.

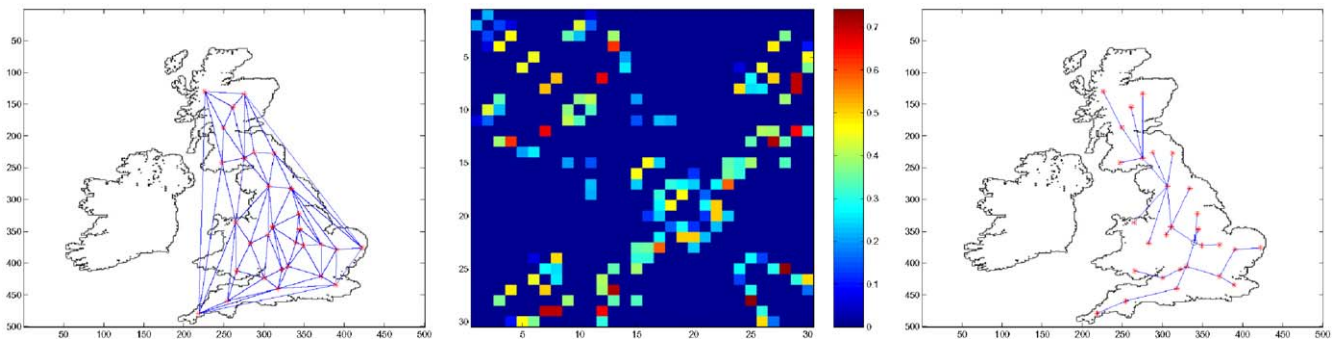


Fig. 4. From left-to-right: Delaunay triangulation corresponding to 30 cities in the UK; corresponding matrix W ; minimum spanning tree delivered by our algorithm.

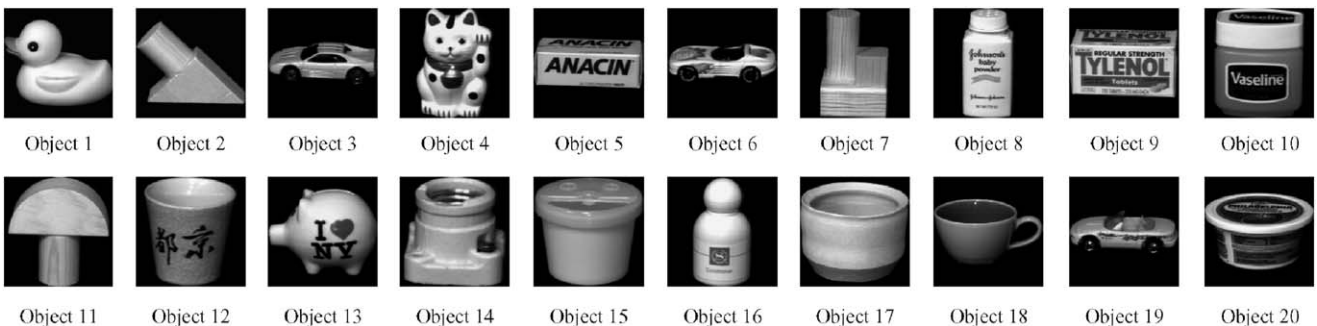


Fig. 5. Sample views for the objects in the Columbia University COIL database.

Table 1

Normalised average ratio ε of incorrect to correct correspondences for the COIL objects as a function of the sectional curvature κ

Object index	Normalised average ratio ε of incorrect to correct correspondences					
	Non-iterative Procrustes alignment			Iterative Procrustes alignment [23]		
	$\kappa = -15$	$\kappa = 0$	$\kappa = 15$	$\kappa = -15$	$\kappa = 0$	$\kappa = 15$
1	0.076	0.079	0.081	0.0699	0.074	0.0775
2	0.078	0.08	0.084	0.0739	0.0776	0.0783
3	0.078	0.083	0.085	0.0717	0.0766	0.0811
4	0.076	0.087	0.091	0.0677	0.0691	0.0867
5	0.088	0.096	0.098	0.0795	0.0837	0.0949
6	0.075	0.089	0.099	0.0652	0.0742	0.0861
7	0.088	0.081	0.086	0.089	0.0886	0.0767
8	0.082	0.087	0.098	0.0673	0.0806	0.0833
9	0.078	0.087	0.089	0.0651	0.0743	0.0856
10	0.081	0.087	0.096	0.0714	0.0782	0.0865
11	0.078	0.085	0.094	0.0563	0.0716	0.0818
12	0.076	0.096	0.098	0.0608	0.0641	0.095
13	0.064	0.084	0.089	0.0357	0.0574	0.0818
14	0.072	0.074	0.081	0.0641	0.071	0.0701
15	0.068	0.076	0.089	0.0539	0.0632	0.068
16	0.071	0.077	0.088	0.0595	0.07	0.0758
17	0.078	0.087	0.087	0.062	0.0705	0.087
18	0.081	0.084	0.085	0.0756	0.0781	0.0832
19	0.086	0.089	0.091	0.083	0.0842	0.0874
20	0.083	0.085	0.087	0.0815	0.0829	0.0834

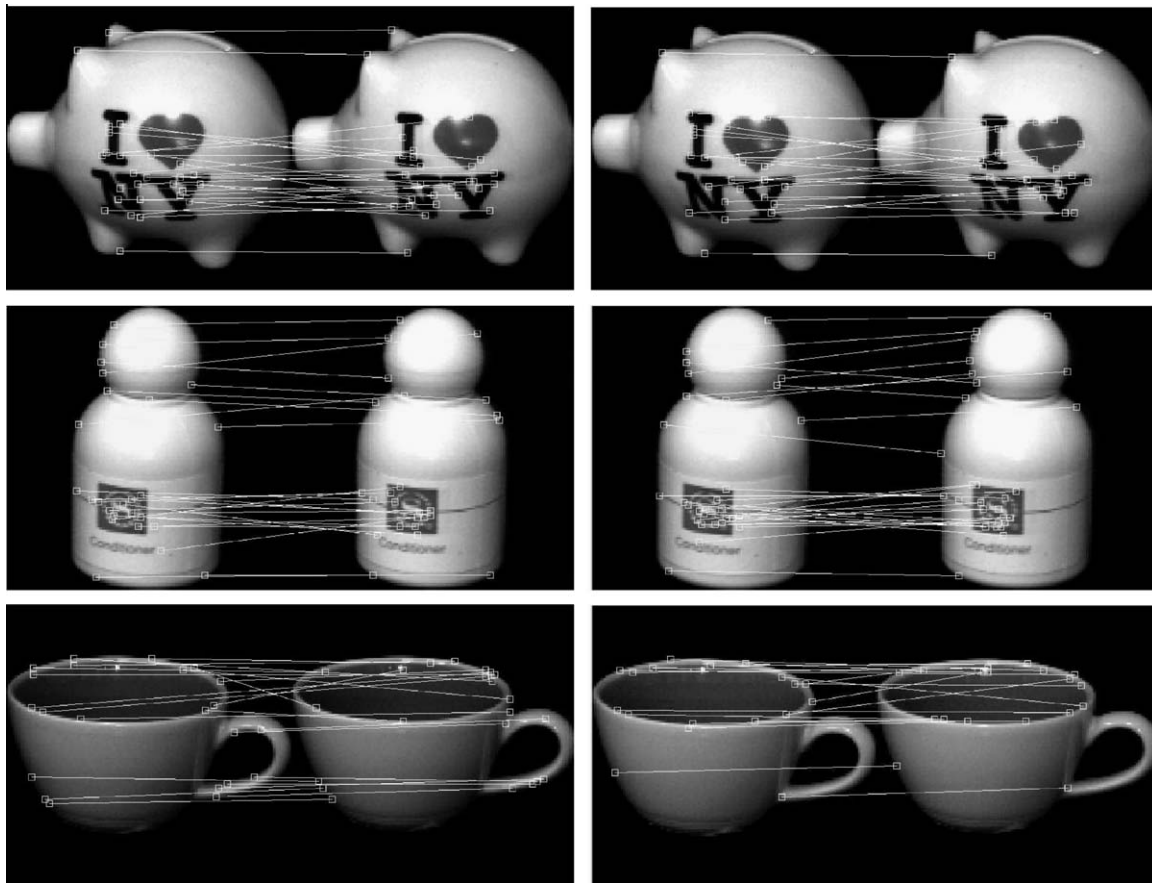
Fig. 6. Sample results, recovered by setting $\kappa = -10$, for three objects in the COIL database. Left-hand column: non-iterative Procrustes alignment; right-hand column: iterative Procrustes alignment.

Table 2
Percentage of correctly recovered data views for each object in the COIL database

Object index	Percentage of correctly recovered data views					
	Non-iterative Procrustes alignment (mean/std. deviation)			Iterative Procrustes alignment [23] (mean/std. deviation)		
	$\kappa = -15$ (%)	$\kappa = 0$ (%)	$\kappa = 15$ (%)	$\kappa = -15$ (%)	$\kappa = 0$ (%)	$\kappa = 15$ (%)
1	95.7/0.309	94.3/0.79	93.9/0.369	97.8/0.179	96.9/0.458	94.3/0.214
2	96.2/0.261	95.2/0.767	94.2/0.561	98.4/0.313	97/0.919	95.4/0.672
3	95.7/0.438	94.8/0.702	93.7/0.718	97.4/1.41	96.3/0.47	95.2/0.6
4	96.3/0.369	94.3/0.968	92.8/1.16	99.3/0.751	97.2/0.97	94.9/0.36
5	96.4/0.679	93.6/0.25	91.6/0.212	101/0.45	96.6/0.79	93.7/0.453
6	94.7/1.19	92.8/0.895	92.1/0.075	98.4/0.31	96.3/0.989	93.5/0.083
7	95.4/1.73	94.7/0.142	93.2/0.813	96.2/0.012	95.5/0.098	94.9/0.056
8	96.4/0.478	95.4/0.58	94.7/0.004	97.8/0.493	96.8/0.598	95.7/0.049
9	95.6/0.45	94.8/0.533	93.6/0.542	95.9/0.74	95.6/0.07	95.2/0.1
10	95.2/2.45	94.5/0.474	92.5/0.391	96.1/1.55	95.4/0.686	94.8/0.566
11	96.4/0.012	96.1/0.283	94.7/1.1	98/0.010	96.7/0.236	97.4/0.92
12	95.8/0.706	95.1/0.539	94.6/0.309	97.2/0.625	96.4/0.477	95.3/0.274
13	96.8/1.68	95.9/0.664	93.7/0.034	97.1/2.31	97.1/0.1	97.5/0.108
14	95.6/1.63	94.9/0.606	93.1/1.6	97.6/0.539	96.2/0.2	95.6/0.529
15	94.9/1	93.8/1.09	91.7/1.6	96.9/0.931	95.4/1.01	94.6/0.48
16	95.2/0.462	94.7/0.252	93.2/1.36	97.3/3.91	95.5/2.13	95.5/0.5
17	96.1/0.707	95.3/0.503	94.4/0.683	97.1/0.269	96.6/0.191	95.5/0.259
18	95.9/0.657	94.4/1.19	93.7/0.267	97/0.067	96.9/0.122	94.8/0.027
19	96.2/1.08	95.9/0.135	94.6/0.43	97.1/0.526	96.4/0.065	96.7/0.21
20	95.8/1.97	94.3/0.787	93.7/0.302	98.9/0.711	96.8/0.284	94.3/0.109

correspondences between each pair of graphs than its non-iterative sibling (Fig. 6).

4.2.2. Retrieval

For our feature-based image retrieval experiments, we have removed 24 out of the 72 views for each object, i.e. every other view. The corners recovered from these views are our model point patterns. The feature points extracted from the remaining views in the database constitute our data set, i.e. 20×48 views. We then match each pair of feature point patterns, corresponding to the “model” and “data” views, so as to retrieve the four images from the data set which amount to the smallest values of the error function ε , as given in Eq. (19). Ideally, these scheme should select the two “data” views indexed immediately before and after the “model” view. In other words, the correct retrieval results for the “model” view indexed i are those views indexed $i - 2, i - 1, i + 1$ and $i + 2$. We use the percentage of correctly recovered views as a measure of the accuracy of the matching algorithm. In Table 2, we show the mean and standard deviation for the percentage of correctly recovered views for both alignment algorithms, i.e. the iterative and the non-iterative one, as a function of the three values of κ and object index. Again, we have used the indexing provided in Fig. 5. The retrieval rate is better for $\kappa = -15$. Recall that, in our matching results, the quantity ε was, for both alternatives, smaller for negative values of κ . The better retrieval rates for $\kappa = -15$ are due to the better quality of the matching results delivered by both algorithms for $\kappa = -15$. This is due to the fact that, as the performance

of the algorithm increases, the value of the quadratic error function ε decreases and, hence, the retrieval rate is higher.

5. Conclusions

In this paper, we have shown how the nodes of a graph can be embedded on a constant sectional curvature manifold. The analysis hinges on the properties of Jacobi fields, and are used to show how to compute an edge-weight matrix in which the elements reflect the sectional curvatures associated with the geodesic paths on the manifold between nodes. To embed the nodes of the graph onto a Riemannian manifold, we apply a doubly centred multidimensional scaling technique to the Laplacian matrix computed from the edge-weights. The embedding coordinates are given by the eigenvectors of the centred Laplacian.

The procedure can be viewed as a transformation to the edge-weights of the graph, which modifies the edge-weights using the sectional curvature. When the sectional curvature is negative, then the effect is to emphasise local or short-distance relationships. When the sectional curvature is negative on the other hand, then the effect is to emphasise long-distance relationships. Using the embedded coordinates corresponding to the nodes of the graph, we show how the problem of graph-matching can be transformed into one of Procrustean point-set alignment.

The work described here can be further extended and improved in a number of different ways. Firstly, there is clearly scope for developing a more sophisticated method for the recovery of the embedding coordinates from the

graph Laplacian. In particular, embedding methods are often used to map data whose pairwise distances are noisy or inaccurate. Hence, the introduction of statistical error measures may be used to improve the results of the algorithm. Likewise, the alignment of the embedding coordinates may also be improved by using of statistical techniques, such as the EM algorithm [23], as an alternative to the Procrustes rotation used here.

References

- [1] C. Gotsman, X. Gu, A. Sheffer, Fundamentals of spherical parameterization for 3D meshes, *ACM Trans. Graphics* 22 (3) (2003) 358–363.
- [2] I. Borg, P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, Springer Series in Statistics, Springer, Berlin, 1997.
- [3] G. Di Battista, P. Eades, R. Tamassia, I. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [4] H. Busemann, *The Geometry of Geodesics*, Academic Press, New York, 1955.
- [5] A. Ranicki, *Algebraic l-theory and Topological Manifolds*, Cambridge University Press, Cambridge, 1992.
- [6] G.R. Hjaltason, H. Samet, Properties of embedding methods for similarity searching in metric spaces, *Pattern Anal. Mach. Intell.* 25 (2003) 530–549.
- [7] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500) (2000) 2319–2323.
- [8] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [9] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: *Neural Information Processing Systems*, vol. 14, 2002, pp. 634–640.
- [10] M. Hein, J. Audibert, U. von Luxburg, From graphs to manifolds—weak and strong pointwise consistency of graph Laplacians, in: *Proceedings of the 18th Conference on Learning Theory*, 2005, pp. 470–485.
- [11] S. Ullman, Filling in the gaps: the shape of subjective contours and a model for their generation, *Biol. Cybern.* 25 (1976) 1–6.
- [12] W.J. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (8) (1995) 749–764.
- [13] S. Gold, A. Rangarajan, A graduated assignment algorithm for graph matching, *Pattern Anal. Mach. Intell.* 18 (4) (1996) 377–388.
- [14] F.R.K. Chung, *Spectral Graph Theory*, American Mathematical Society, Providence, RI, 1997.
- [15] S. Umeyama, An eigen decomposition approach to weighted graph matching problems, *Pattern Anal. Mach. Intell.* 10 (5) (1988) 695–703.
- [16] G. Scott, H. Longuet-Higgins, An algorithm for associating the features of two images, *Proc. R. Soc. London Ser. B* 244 (1991) 21–26.
- [17] L. Shapiro, J.M. Brady, Feature-based correspondence—an eigenvector approach, *Image Vision Comput.* 10 (1992) 283–288.
- [18] B. Luo, E.R. Hancock, Structural graph matching using the EM algorithm and singular value decomposition, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (10) (2001) 1120–1136.
- [19] T.M. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching, *Pattern Anal. Mach. Intell.* 26 (4) (2004) 515–519.
- [20] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models—their training and application, *Comput. Vision Image Understanding* 61 (1) (1995) 38–59.
- [21] P. Torr, D.W. Murray, The development and comparison of robust methods for estimating the fundamental matrix, *Int. J. Comput. Vision* 24 (1997) 271–300.
- [22] H. Le, C.G. Small, Multidimensional scaling of simplex shapes, *Pattern Recognition* 32 (9) (1999) 1601–1613.
- [23] B. Luo, E. Hancock, Iterative procrustes alignment with the EM algorithm, *Image Vision Comput.* 20 (2002) 377–396.
- [24] R. Tomassia, On-line planar graph embedding, *J. Algorithms* 21 (2) (1996) 201–239.
- [25] A. Papakostas, I. Tollis, Efficient orthogonal drawings of high degree graphs, *Algorithmica* 26 (1) (2000) 100–125.
- [26] I. Chavel, *Riemannian Geometry: A Modern Introduction*, Cambridge University Press, Cambridge, 1995.
- [27] I. Chavel, *Eigenvalues in Riemannian Geometry*, Academic Press, New York, 1984.
- [28] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 888–905.
- [29] S. Sarkar, K.L. Boyer, Quantitative measures of change based on feature organization: eigenvalues and eigenvectors, *Comput. Vision Image Understanding* 71 (1) (1998) 110–136.
- [30] J. Lafferty, G. Lebanon, Diffusion kernels on statistical manifolds, *J. Mach. Learn. Res.* 6 (2005) 129–163.
- [31] W.S. Torgerson, Multidimensional scaling I: theory and method, *Psychometrika* 17 (1952) 401–419.
- [32] G. Young, A.S. Householder, Discussion of a set of points in terms of their mutual distances, *Psychometrika* 3 (1938) 19–22.
- [33] M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [34] H. Flanders, *Differential Forms with Applications to the Physical Sciences*, Dover, New York, 1990.
- [35] B. O’Neill, *Elementary Differential Geometry*, Academic Press, New York, 1997.
- [36] M. Berger, *A Panoramic View of Riemannian Geometry*, Springer, Berlin, 2003.
- [37] U. von Luxburg, O. Bousquet, M. Belkin, Limits of spectral clustering, in: *Advances in Neural Information Processing Systems*, vol. 17, 2005, pp. 857–864.
- [38] S. Zucker, Estimates for the classical parametrization for the Laplacian, *Manuscripta Mathematica* 24 (1) (1978) 1432–1785.
- [39] M. Belkin, P. Niyogi, Towards a theoretical foundation for Laplacian-based manifold methods, in: *Proceedings of the 18th Conference on Learning Theory*, 2005, pp. 486–500.
- [40] A. Singer, From graph to manifold Laplacian: the convergence rate, *Appl. Comput. Harmonic Anal.* 21 (1) (2006) 128–134.
- [41] J. McCleary, *Geometry from a Differentiable Viewpoint*, Cambridge University Press, Cambridge, 1997.
- [42] S. Rosenberg, *The Laplacian on a Riemannian Manifold*, Cambridge University Press, Cambridge, 1997.
- [43] J. Jost, *Riemannian Geometry and Geometric Analysis*, Springer, Berlin, 2002.
- [44] E.R. Hancock, J.V. Kittler, Discrete relaxation, *Pattern Recognition* 23 (1990) 711–733.
- [45] C.J. Harris, M. Stephens, A combined corner and edge detector, in: *Proceedings of the Fourth Alvey Vision Conference*, 1988, pp. 147–151.

About the Author—ANTONIO ROBLES-KELLY received his B.Eng. degree in Electronics and Telecommunications from the Inst. Tecnológico y de Estudios Superiores de Monterrey with honours in 1998. In 2001, being a graduate student at York, he visited the University of South Florida as part of the William Gibbs/Plessey Award to the best research proposal to visit an overseas research lab. He received his Ph.D. in Computer Science from the University of York in 2003. After receiving his doctorate, Dr. Robles-Kelly remained in York until December 2004 as a Research Associate under the MathFit-EPSRC framework. Currently, he is a research scientist with National ICT Australia (NICTA) at the Canberra Lab, where he leads the Spectral Imaging and Source Mapping project of the Vision Systems, Technologies and Applications programme.

His research interests are in the areas of Computer Vision, Pattern Recognition and Computer Graphics. Along these lines, he has done work on segmentation and grouping, graph-matching, shape-from-X and reflectance models. He is also interested in the differential structure of surfaces.

About the Author—EDWIN R. HANCOCK studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high energy physics in 1981. Following this he worked for 10 years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period, he also held adjunct teaching posts at the University of Surrey and the Open University. In 1991, he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer in 1997 and to Reader in 1998. In 1998, he was appointed to a Chair in Computer Vision.

Professor Hancock now leads a group of some 15 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimisation and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image data-bases and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing and medical imaging. Professor Hancock has published some 100 journal papers and 400 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 and an outstanding paper award in 1997 by the journal *Pattern Recognition*. In 1998, he became a fellow of the International Association for Pattern Recognition.

Professor Hancock has been a member of the Editorial Boards of the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*. He has also been a guest editor for special editions of the journals *Image and Vision Computing* and *Pattern Recognition*. He has been on the programme committees for numerous national and international meetings. In 1997 with Marcello Pelillo, he established a new series of international meetings on energy minimisation methods in computer vision and pattern recognition.