Nonlinear Dimensionality Reduction for Kinematic Cartography with an Application Toward Robotic Locomotion

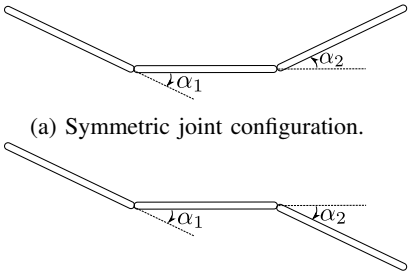
Tony Dear[1], Ross L. Hatton[2], and Howie Choset[1]

***Abstract*— Planning robot motions often requires a notion of the "distance" between configurations or the "length" of a trajectory connecting them in the configuration space. If these quantities are defined so as to correspond to the effort required to change configurations, then they would likely differ from the Euclidean distance or arclength in the system's configuration parameters, distorting the visual representation of the relative costs of executing the motions. This problem is fundamentally similar to that of producing map projections with minimal distortion in cartography. A separate problem is that of nonlinear dimensionality reduction (NLDR), which, given a set of data, projects it into a lower-dimensional space while seeking to retain the geometric relationship between data points. In this paper, we show that NLDR can be applied to the kinematic cartography problem, allowing us to generate system parameterizations in which distance and arclength correspond to the effort of motion.**

I. Introduction

A natural evaluation criterion for robot motion planning is the "distance" between two configurations or the "length" of a trajectory. These quantities may be computed as the Euclidean distance between the parameters identifying the two configurations, or as the arclength of the trajectory according to the parameters. However, these measures can misrepresent the effort required to achieve the motion. For example, it takes considerably more effort for a three-link swimmer, starting from a straight configuration ($\alpha_1 = \alpha_2 = 0$) and immersed in a viscous fluid, to achieve the symmetric shape in Fig. 1a than it does to achieve the antisymmetric shape in Fig. 1b. As explained in [1], the fluid drag forces on the links result in the joint motors working against each other in the symmetric mode, but working together in the antisymmetric case. This difference is not apparent in the swimmer's configuration space, where both configurations have the same Euclidean distance from its starting configuration.

Such discrepancies in effort can often be encoded in a *Riemannian metric* that weights the length of each differential configuration change according to its direction and its location in the configuration space. The length of a trajectory through the space may be integrated up from the weighted differential distances, while the distance between two points can be defined as the length of the shortest trajectory connecting them. In this context, a question naturally arises: "What parameterizations of a system best capture the structure of its configuration space, minimizing the discrepancy between Euclidean distances and arclengths on the parameters and the true distances and lengths according to the metric?" Such parameterizations are then useful for graphically representing the system, as they inherently represent expensive motions as visually longer than those that are easier to execute.



(a) Symmetric joint configuration.

(b) Antisymmetric joint configuration.

Fig. 1: Two joint configurations for the three-link swimmer.

In [1], the authors addressed this question by generalizing the cartographic problem of identifying map projections of the globe with minimal distortion. Here the true metric distances are those traveled over the earth's surface, which are not well-represented by distances in latitude-longitude parameterizations. Drawing on work from the cartographic community [2], [3], the authors of [1] were able to characterize the distortion of a metric under a given parameterization and develop an algorithm for finding new parameterizations to minimize the distortion. The latter was achieved by treating the parameter space as a pre-strained elastic sheet and relaxing it to a minimum-energy configuration; this approach produced demonstrably useful results, but had no formal guarantees of convergence or stability.

In this paper, we explore an alternative means of optimizing system parameterizations, based on techniques from nonlinear data reduction (NLDR) [4]–[12]. These techniques help to identify low-dimensional structures embedded within higher-dimensional data sets—a classic example is picking out the spiral sheet that describes the position of points in the "Swiss roll" data set in Fig. 2. In doing so, NLDR algorithms can be thought of as generalizations of principal component analysis (PCA) or multidimensional scaling (MDS) [13] of the data set, techniques that would only be able to identify linear structures in the data—for example, a best-fit plane rather than the spiral. Once the lower-dimensional structure within a data set is found, it can be used to project the data set into a space of that dimension, for example "unrolling" the data set in Fig. 2 into a flat ribbon.

[1]T. Dear and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA. {tonydear@, choset@cs.}cmu.edu

[2]R. L. Hatton is with the School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331, USA. ross.hatton@oregonstate.edu

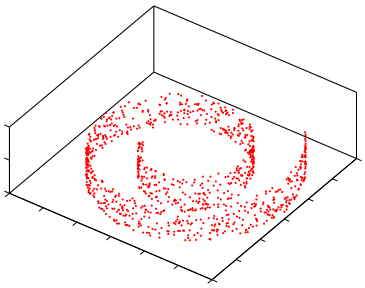


Fig. 2: The "Swiss roll" data set, which largely lie on a rolled up two-dimensional manifold embedded within 3-space.

One such algorithm, Isomap [4], identifies the low-dimensional data structure and makes its projection by first connecting neighboring nodes into a graph whose edge-lengths are the Euclidean distances between the high-dimensional locations of the points. Isomap then defines the distance between non-neighboring nodes as the shortest path connecting them in the graph, and then projects the points onto the lower-dimensional output space by performing MDS on the full matrix of point-to-point distances.

The first stage of this process is essentially a discrete computation of inter-point distances in the presence of a Riemannian metric, the only difference being that Euclidean distances in a higher dimensional space are considered rather than metric-weighted distances. We therefore propose that a modified implementation of Isomap can achieve our goal of optimally parameterizing a robot's configuration space, as well as carry over the algorithm's formal convergence properties. To justify this proposition, we apply Isomap to project the swimming system considered in [1] and the globe into minimum-distortion coordinates. We then compare these results to those from our earlier elastic-relaxation algorithm, and discuss the relative advantages of each approach.

II. Kinematics and Cartographic Distortion

We briefly review the background describing the relationship between locomotion kinematics and cartographical analysis. For a fuller treatment of the subject, see [1].

A. The Geometric Mechanics of Locomotion

The configuration space Q of a locomoting system can be separated into a position space G and a shape manifold M, such that the position $g \in G$ locates the system in the world, and the shape $r \in M$ describes the relative positions of the particles in the system. For example, the position coordinates of the three-link system in Fig. 3, $g = (x, y, \theta) \in SE(2)$, describe the position of the center of mass of the center link and the orientation of the same; the shape variables $r = (\alpha_1, \alpha_2) \in R^1 \times R^1$ describe its two joint angles.

Locomotion is the means by which changes in shape affect the position of the system. Through the use of techniques from geometric mechanics [14]–[19], this process is often characterized by differential equations that relate the *body velocity* ξ to the shape velocity $\dot{r}$ and the accumulated momentum of the system. This body velocity is the rate

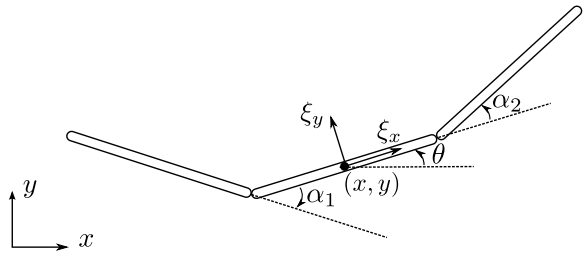


Fig. 3: Geometry of the three-link swimmer.

of change of the position variables g expressed in a body-attached frame, as shown in Fig. 3.

For systems that are sufficiently constrained, the generalized momentum drops out and the system behavior is dictated by the *kinematic reconstruction equation*,

$$\xi = -\mathbf{A}(r)\dot{r}, \tag{1}$$

in which the *local connection* $\mathbf{A}(r)$ acts as a Jacobian [15], mapping shape velocities to body velocities. The local connection can be used to describe a range of systems—a key class of systems includes those in low Reynolds number fluids, in which a linear shape velocity to body velocity relationship is dictated by linear fluid drag and a quasi-static force-balancing assumption on the system. The details of the computation of the three-link swimmer's local connection may be found in [1] and [20].

In addition to encoding the differential kinematic relationship between shape and position changes, the structure of the local connection also contains information about the system's locomotive capabilities over longer motions. The net displacement corresponding to a gait, or closed trajectory in the shape space, can be approximated by the area integral of the local connection's curvature over the shape space region enclosed by the gait. Historically this approximation was limited to small motions, but recent work has shown that it can be made more accurate over a large range of gaits by incorporating an auxiliary coordinate optimization [21].

Intuitively, the "cost" to execute a gait is related to the path length of the gait in the system's shape space. The difficulty of using this measure arises when the shape space is not homogeneous; *i.e.*, joint costs may not be independent of each other, and costs of changing shape may depend on the current shape configuration, as in Fig. 1.

B. Distance Metrics in the Shape Space

The *distance metric* function on a space gives the distance between two elements of that space; the *metric tensor* is a coordinate representation of a metric function. For example, in two-dimensional Euclidean space, the differential arclength or line element $ds_{\mathbb{R}}$, corresponding to differential displacements in x and y, may be defined by

$$ds^2_{\mathbb{R}} = dx^2 + dy^2 = \begin{bmatrix} dx & dy \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathcal{M}_{\mathbb{R}}} \begin{bmatrix} dx \\ dy \end{bmatrix}, \tag{2}$$

where $\mathcal{M}_{\mathbb{R}}$ is the metric tensor corresponding to the Euclidean metric and $ds^2_{\mathbb{R}}$ is the *first fundamental form*. We can then integrate the line element $ds_{\mathbb{R}}$ to find the length of a trajectory between two points. In general, the metric tensor need not be the identity and may reflect different contributions in distance by each of the components.

For mechanical systems, the length of trajectories in the shape space can be defined similarly. For example, for the three-link swimmer of Fig. 3 with two joint angles, the first fundamental form can be written as

$$ds^2_\alpha = \begin{bmatrix} d\alpha_1 & d\alpha_2 \end{bmatrix} \mathcal{M}_\alpha \begin{bmatrix} d\alpha_1 \\ d\alpha_2 \end{bmatrix}, \tag{3}$$

where $\mathcal{M}_\alpha$ is the corresponding shape metric tensor. If $\mathcal{M}_\alpha$ is the identity, then this metric would be the Euclidean metric on the space of joint angles, which does not encode any notion of "effort" in producing motion.

The authors of [1] explicitly use the dissipation forces of low Reynolds swimming to find a configuration metric, such that arclength in the shape space measures the effort taken and minimum-effort changes between shape correspond to *geodesics*, or shortest path distances. To derive this metric, the power p dissipated through the joints may be written as a quadratic function of the joint velocities, *i.e.*,

$$p = \dot{\alpha}^T f(\alpha) \dot{\alpha} = \left(\frac{d\alpha}{dt}\right)^T f(\alpha) \left(\frac{d\alpha}{dt}\right). \tag{4}$$

Here $f(\alpha)$ incorporates the kinematics relating joint velocities to link velocities, as well as the drag forces on the links. Multiplying both sides by dt^2, the function $f(\alpha)$ then takes on the role of the metric tensor, *i.e.*,

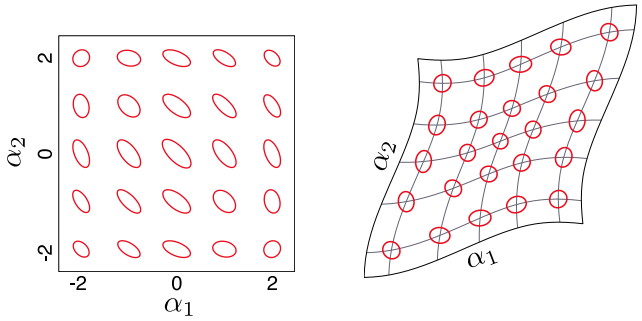
$$ds^2_p = p\, dt^2 = \begin{bmatrix} d\alpha_1 & d\alpha_2 \end{bmatrix} f(\alpha) \begin{bmatrix} d\alpha_1 \\ d\alpha_2 \end{bmatrix}, \tag{5}$$

where the line element is defined as $ds_p = \sqrt{p}\, dt$. We can thus define the "cost" of executing a gait based on the amount of power required to execute it.

C. Visualization of Distances

In addition to encoding the costs to move from one configuration to another, the metric tensor also provides a means for distance visualization by indicating the amount of metric distortion, due to the chosen parameterization, present at and between various points on the shape manifold. Configurations at which the tensor has large deviation from the scaled identity indicate large distortion in how the manifold is displayed on a Euclidean surface, while changes in metric tensor scaling indicate that different points are unequally weighted in the chosen coordinate representation.

For example, such distortion is encountered in equirectangular map projections in geographic cartography, as straight lines on the map are not geodesics and different regions of the map are stretched out to different proportions. Similarly for the configuration spaces of many systems, the standard Euclidean grid representation of the shape manifold misrepresents the relationship between effort and path length. Hence, one solution for rectifying this mismatch between parameterization and true effort would be to "stretch out" the current paramterization of the shape space to one that better represents distances on a flat grid.



(a) Original coordinates. (b) Normalized coordinates.

Fig. 4: Tissot's indicatrices for the three-link swimmer's power dissipation metric.

In [1], the authors implement this idea through a spring relaxation algorithm. Shape space points are first evenly sampled in the chosen coordinates. Springs are then attached to pairs of neighboring nodes, where their initial lengths are determined by the Euclidean metric on the current coordinate representation, and their neutral lengths computed according to the distance metric. These springs are thus pre-loaded with a strain given by the difference between their true "metric" distances and their Euclidean lengths on a two-dimensional surface. After the initial setup, the springs are allowed to iteratively relax into a minimum-energy configuration.

As the springs relax, the attached starting points are pulled into new positions on the plane, and inter-point distances remain or become more consistent with the distance metric. The new point coordinates then give a better basis for the space, and the metric tensor in this new basis is given by

$$\mathcal{M}_{\text{new}} = J^{-T} \mathcal{M}_{\text{original}} J^{-1}, \tag{6}$$

where J is the Jacobian of the transformation function mapping from the original node positions to final positions.

Figure 4 shows the result of applying spring relaxation to the shape space of the three-link swimmer parameterized by its joint angles. We start with a standard Euclidean grid for the shape space in Fig. 4a, where points represent swimmer configurations. The power metric from Eq. (5), however, shows that this representation is quite distorted; the ellipses are the *Tissot's indicatrices* [1], [2], where the amount of elongation in one direction at each point corresponds to the "stretch" of the manifold in that direction at the point.

Once the swimmer's shape manifold is reparameterized in a minimum-energy configuration in Fig. 4b, the indicatrices are more circular, and Euclidean distances more accurately represent the effort of motion. In particular, points along the $(+\alpha_1, -\alpha_2)$ diagonal become closer together than those along the $(+\alpha_1, +\alpha_2)$ diagonal. For a given number of radians, it is cheaper to change angles in the asymmetric direction than it is to do so along the symmetric direction, confirming the observation from Fig. 1.

III. Isomap for Kinematic Cartography

The spring relaxation method optimizes the projection so that every segment of unit differential length according to the metric corresponds as closely as possible to a segment of unit differential length on a flat grid. By minimizing metric distortion, it ensures that Euclidean distances in the new parameterization correspond well to actual distances on the manifold determined by the distance metric.

This objective is similar to that used by the original Isomap algorithm [4] projecting data into a two-dimensional space, with the objective of ensuring that their resultant inter-point distances correspond to the geodesics of the underlying geometric structure. However, while the input to the traditional Isomap algorithm is a set of data points, we start with a metric tensor defined throughout the shape space.

In the first step, we uniformly sample n discrete points from our domain of interest and form a flat k-connected graph H. For example, if $k = 8$ then most nodes will be connected to the k nearest neighbors in the cardinal and intercardinal directions, while domain boundary nodes may be connected to fewer neighbors. These nodes will form the dataset to be input into the original Isomap algorithm. The sample size n and connectivity k both dictate the tradeoff between accuracy and runtime in the algorithm.

We next define $d_X(i, j)$ as the edge length between nodes i and j; we compute this for all existing edges in H by passing the respective node-to-node vector through the metric tensor, producing the neutral spring length as described in [1]. We emphasize that these estimated distances only approximate the true geodesic distances, $d_T(i, j)$, well if the sampling density is sufficiently high, as they are computed by integration of the infinitesimal metric tensors at the nodes.

The remaining steps are identical to those of the original Isomap algorithm—finding the matrix of estimated graph distances D_H and then applying classical multidimensional scaling (MDS) [22], [23]. To do so, we use our approximations $d_X(i, j)$ and compute the remaining distances between non-adjacent nodes using standard methods for finding the shortest path distances, such as the Dijkstra or Floyd-Warshall algorithms. We thus end up with a full distance matrix D_H as an approximation of the geodesic matrix D_M.

Finally, an embedding of the graph in d-dimensional Euclidean space (for our goal of distance visualization, $d = 2$) can be found by applying classical MDS to D_H. The first step is to center D_H by defining the inner product matrix

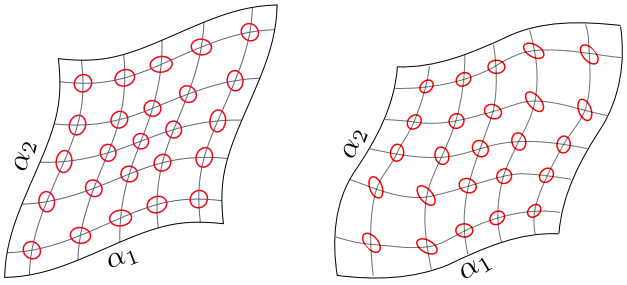
$$\mathcal{IP}(D_H) = -\frac{1}{2} C_n (D_H)^2 C_n, \tag{7}$$

where $(D_H)^2$ is the entrywise squared distance matrix and

$$C_n = I_n - \frac{1}{n}\mathbb{O}_n \tag{8}$$

is the $n \times n$ centering matrix, defined as the difference between the identity matrix and the matrix with all elements equal to $\frac{1}{n}$. This is analogous to the data centering step of PCA, so the rows and columns of $\mathcal{IP}(D_H)$ sum to 0.

We then perform an eigendecomposition of $\mathcal{IP}(D_H)$, and the new coordinates of node i are given by $(\sigma_1 v_1^i, \ldots, \sigma_d v_d^i)$, where σ_1 through σ_d are the largest d singular values, and v_p^i is the ith element of the eigenvector corresponding to the pth singular value. The output is thus a set of n separate d-dimensional coordinate vectors, each describing the new location of each of the original nodes. These vectors give a new parameterization of the space, which fills the same role as the minimum-energy result of spring relaxation.



(a) Spring relaxation. (b) Isomap.

Fig. 5: Comparison of the new parameterization plots based on the power metric for the three-link swimmer.

IV. Results and Discussion

We now assess the performance of Isomap and note its differences from the spring relaxation algorithm along several optimization criteria. We will also use proven results from the NLDR literature to arrive at some theoretical guarantees for Isomap as applied to kinematic cartography.

A. The Three-Link Swimmer

Figure 5 compares the parameterization of the three-link swimmer due to spring relaxation (from Fig. 4b) to that of Isomap side-by-side, with $k = 8$ for both. They are visually quite similar, particularly with the main stretch in the $(+\alpha_1, +\alpha_2)$ direction. However, the Isomap representation appears to retain more of its original local distortion than the former, which is apparent in the distorted indicatrices. This is because the Isomap algorithm optimizes the distances between all pairs of points, not just neighboring points.

Two types of linear distance criteria [3] are considered in [1] for evaluation of the spring relaxation algorithm. The *Tissot criterion* $\text{crit}_T = \tau_{1,\max}/\tau_{2,\min}$ is the ratio between the largest and smallest indicatrix axes in the domain, where the *principal stretches* τ_1 and τ_2 of each indicatrix are the inverse squares of the local metric tensor's singular values.

The other measure is the *Airy criterion*, which measures the average squared stretch of the indicatrices over the domain. Its computation details are given in [1]. The springs tend toward a minimum energy configuration, and strain energy is proportional to the squared stretch of the springs, so the spring relaxation algorithm can be shown to minimize the Airy criterion, specifically its Jordan form crit_J.

On the other hand, our version of the Isomap algorithm minimizes the *residual variance* RV_H [4], defined as

$$RV_H(D_Y) = \|\mathcal{IP}(D_H) - \mathcal{IP}(D_Y)\|_{L_2}. \tag{9}$$

TABLE I: Distortion metrics for the three-link swimmer.

Parameterization	crit_T	crit_J	RV_H
Joint angles	3.13	6.18×10^{-2}	19.0×10^{-2}
Power-normed, springs	1.49	7.85×10^{-3}	1.79×10^{-2}
Power-normed, Isomap	2.16	3.42×10^{-2}	1.25×10^{-2}

D_H is the matrix of metric distance approximations between the sampled nodes, D_Y is a matrix of Euclidean distances of the same nodes in a given parameterization, and $\|\cdot\|_{L_2}$ is the L^2 matrix norm. Physically, RV_H is a measure of how well the Euclidean distances from a given parameterization match the corresponding metric distance approximations.

The difference between the outputs of spring relaxation and Isomap is due primarily to the structure of the computed distance matrix D_H. In the former, each node only considers its immediate k neighbors, while Isomap estimates the remaining internode distances using shortest paths. Isomap can thus be viewed as generalizing spring relaxation by fully connecting the node graph, giving better global accuracy at the expense of lower local accuracy. In future work, we plan to verify this claim by weighting local edges more heavily in our Isomap implementation.

Table I shows the distortion metrics of three different parameterizations of the three-link swimmer shape manifold. The Isomap parameterization ends up with a lower residual variance than the spring relaxation parameterization, while its Tissot and Jordan criteria, both local distortion metrics, are higher. Meanwhile, all three criteria are higher for the original joint-angle parameterization (Fig. 4a) than for the power-normed parameterizations.

B. Convergence Properties

As our implementation of Isomap relies on sampling, it is desirable to establish convergence properties that guarantee its usefulness. As proven in [4], [24], Isomap is guaranteed to asymptotically recover the underlying geometric structure of the sampled points and estimated distances D_H in the lower dimensional space. This property is directly due to the minimization of the residual variance (9).

Summarizing the main results of [24], it can be shown that, for any two points r_1, r_2 in the shape manifold M, the Isomap output $d_Y(r_1, r_2)$ can be bounded both above and below by a factor of the true underlying geodesics $d_T(r_1, r_2)$:

$$(1 - \lambda_1)d_T \leq d_Y \leq (1 + \lambda_2)d_T. \tag{10}$$

Roughly speaking, the factor λ_1 scales with the maximum edge length in D_H, while λ_2 scales inversely with the edge coverage as well as the sampling density of the underlying manifold. Assuming that the size of the shape domain remains constant, these constants can be made arbitrarily small, and thus the bounds arbitrarily tight, by increasing the sample size n and connectivity k appropriately.

To see that the aforementioned results on convergence hold in practice, it would not be very revealing to simply increase the sampling rate of the three-link swimmer shape manifold, as we have no means of quantifying its convergence to the optimal projection. The residual variance (9) only compares the results to the sampled points, and we do not have the true manifold structure, or ground truth.

TABLE II: Distortion metrics for the quarter-sphere.

Parameterization	n	crit_T	crit_J	RV_T
Equirectangular	377	14.1	4.04×10^{-2}	39.6×10^{-2}
Conical, springs	377	2.85	2.63×10^{-3}	1.50×10^{-2}
Conical, Isomap	377	3.93	7.17×10^{-3}	1.23×10^{-2}
Equirectangular	629	14.1	4.21×10^{-2}	38.6×10^{-2}
Conical, springs	629	2.28	2.14×10^{-3}	8.27×10^{-3}
Conical, Isomap	629	3.23	7.66×10^{-3}	6.67×10^{-3}
Equirectangular	1325	14.1	4.35×10^{-2}	37.6×10^{-2}
Conical, springs	1325	1.91	1.76×10^{-3}	5.98×10^{-3}
Conical, Isomap	1325	2.66	7.88×10^{-3}	4.78×10^{-3}

C. Validation of Convergence

For an example in which we can perform a ground truth comparison, we investigate the unit sphere $\mathbb{S}^2$, as we can explicitly compute geodesics on this manifold. The unit sphere has long been studied by cartographers for the purposes of map-making for the earth. It is well-known that equirectangular representations of the earth result in significant distortion near the poles; conical projections, on the other hand, ensure that two-dimensional Euclidean distances are more uniform throughout the map.

Parameterizing the unit sphere by longitude λ and latitude ϕ, the first fundamental form and metric tensor are given by

$$ds^2_{\mathbb{S}} = \begin{bmatrix} d\lambda & d\phi \end{bmatrix} \underbrace{\begin{bmatrix} \cos^2(\phi) & 0 \\ 0 & 1 \end{bmatrix}}_{\mathcal{M}_{\mathbb{S}}} \begin{bmatrix} d\lambda \\ d\phi \end{bmatrix}. \tag{11}$$

The geodesic between two points minimizes $\int ds_{\mathbb{S}}$ over all paths between them, which is the great circle distance.

Because we can compute the geodesics between any pair of points, we can quantify Isomap's results based on the residual variance comparing the projection to the true geometric structure instead of that of the sampled points:

$$RV_T(D_Y) = \|\mathcal{IP}(D_T) - \mathcal{IP}(D_Y)\|_{L_2}. \tag{12}$$

Note that we are simply introducing an alternative means of interpreting the results, rather than changing the underlying algorithm. While D_H does not appear in (12), the sampling density still influences the accuracy of the result D_Y.

Table II presents the distortion metrics for a quarter-sphere for both springs and Isomap with three different sampling densities while keeping $k = 8$. As with the three-link swimmer, the original equirectangular representations have higher local criteria values, while the new representations have lower values, with the springs algorithm performing slightly better. In addition, the residual variance RV_T is consistently lower for Isomap than for springs. Finally, a comparison of RV_T between the three different sample sizes

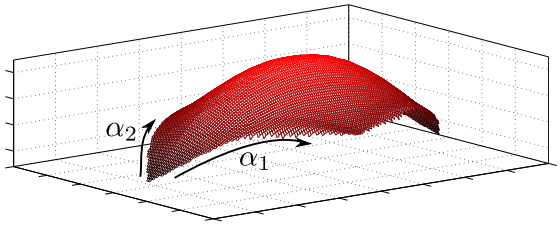


Fig. 6: The power-normalized shape space manifold for the three-link swimmer in three dimensions.

reveals that we do get better results as n increases, which agrees with our convergence arguments.

V. Conclusions and Future Work

In exploring the application of nonlinear dimensionality reduction to kinematic cartography, we have shown one way in which computational tools traditionally employed by two disjoint areas of robotics can be combined. This work has allowed us to add rigor to the spring relaxation algorithm developed in [1], as well as approach the motion planning problem from a manifold learning perspective.

Future work may include a more detailed investigation of how Isomap generalizes the spring algorithm, possibly by reweighting the edges in the constructed graph such that only local neighbors are considered. This modification would match the setup of the spring algorithm, allowing us to perform a more rigorous analysis of the relationship between the linear distance criteria and the residual variance.

In addition to introducing weights, it would be desirable to further examine the effects of varying the sampling density and graph connectivity, so as to rigorously carry over the convergence proofs from [24] to our Isomap implementation. We also envision trying alternative optimization criteria for the underlying MDS algorithm, namely the *stress* criterion that generalizes residual variance [22], [23].

In this paper, we have focused exclusively on projecting manifolds into two dimensions, but the original Isomap algorithm is not limited to $d = 2$. Figure 6 shows Isomap's projection of the three-link swimmer's shape manifold in three dimensions, revealing a significant component off the plane. This is because the curvature of the manifold is related to the derivative of the distance metric. We may thus be able to glean new insights by investigating these projections and their geometric structures in higher dimensions.

Finally, the NLDR literature includes many techniques other than Isomap, such as local linear embedding [5], Laplacian eigenmaps [7], and diffusion maps [28]. Alongside future work in developing Isomap, parallel work may more extensively survey the field and investigate whether alternative techniques can be applied to this problem to yield better or more revealing results.

References

[1] R. L. Hatton and H. Choset, "Kinematic cartography for locomotion at low reynolds numbers," *Proc. Robot. Sci. Syst. VII*, 2011.

[2] A. Tissot, *Mémoire sur la représentation des surfaces et les projections des cartes géographiques*. Gauthier-Villars, 1881.

[3] A. Niermann, "A comparison of various advantageous cartographical representations," Master's thesis, Universität Stuttgart, Stuttgart, Germany, June 1983.

[4] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[6] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in neural information processing systems*, 2002, pp. 705–712.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[8] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[9] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 47.

[10] J. J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer, 2007.

[11] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 40–51, 2007.

[12] L. Van der Maaten, E. Postma, and H. Van Den Herik, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, vol. 10, pp. 1–41, 2009.

[13] K. V. Mardia, J. T. Kent, and J. M. Bibby, "Multivariate analysis," 1980.

[14] A. M. Bloch, *Nonholonomic mechanics and control*. Springer, 2003, vol. 24.

[15] R. L. Hatton and H. Choset, "Optimizing coordinate choice for locomoting systems," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4493–4498.

[16] F. Wilczek and A. Shapere, *Geometric phases in physics*. World Scientific, 1989, vol. 5.

[17] S. D. Kelly and R. M. Murray, "Geometric phases and robotic locomotion," *Journal of Robotic Systems*, vol. 12, no. 6, pp. 417–431, 1995.

[18] J. Ostrowski and J. Burdick, "The geometric mechanics of undulatory robotic locomotion," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 683–701, 1998.

[19] E. A. Shammas, H. Choset, and A. A. Rizzi, "Geometric motion planning analysis for two classes of underactuated mechanical systems," *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1043–1073, 2007.

[20] R. L. Hatton and H. Choset, "Connection vector fields and optimized coordinates for swimming systems at low and high reynolds numbers," in *Proceedings of the ASME Dynamic Systems and Controls Conference (DSCC), Cambridge, Massachusetts, USA*, 2010.

[21] ——, "Geometric motion planning: The local connection, stokes theorem, and the importance of coordinate choice," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 988–1014, 2011.

[22] I. Borg, *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[23] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2010.

[24] M. Bernstein, V. De Silva, J. C. Langford, and J. B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," Technical report, Department of Psychology, Stanford University, Tech. Rep., 2000.

[25] M. Belkin and P. Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," in *Learning theory*. Springer, 2005, pp. 486–500.

[26] ——, "Convergence of laplacian eigenmaps," *Advances in Neural Information Processing Systems*, vol. 19, p. 129, 2007.

[27] M. Hein, J.-Y. Audibert, and U. Von Luxburg, "From graphs to manifolds–weak and strong pointwise consistency of graph laplacians," in *Learning theory*. Springer, 2005, pp. 470–485.

[28] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.