1

Variants 1–2

In this chapter, we introduce the classical Whitney extension problem. Thereafter, we introduce the near distorted Whitney extension problem and two variants of it. The first, via a purely harmonic analysis problem and the second, translated into a problem related to non-rigid alignment and interpolation of data in $\mathbb{R}^d$. We discuss the Procrustes rigid alignment problem.

COPYRIGHTED MATERIAL

1.1 The Whitney Extension Problem

Given a real valued function ϕ on an arbitrary compact set in $\mathbb{R}^n$, the classical Whitney extension problem asks how can one decide whether ϕ extends to a function Φ in $C^m(\mathbb{R}^n)$, $m \geq 1$, the space of real valued functions on $\mathbb{R}^n$ whose derivatives of order m are continuous and bounded? Whitney [114, 115] first studied this problem in 1934. He solved the real line case ($n = 1$) and proved the classic Whitney extension theorem. See [63, 64] and the references cited therein for an interesting account of this problem.

1.2 Variants (1–2) [39, 40]

Problems (1–2) are examples of Variants (1–2).

Problem 1. *Let us be given a positive constant c small enough depending on d. Does there exist a positive constant c′ small enough depending on c so that the following holds? Given two sets of* $k \geq 1$ *distinct points in* $\mathbb{R}^d$, $\{y_1, \ldots, y_k\}$ *and* $\{z_1, \ldots, z_k\}$. *Suppose for every* $1 \leq i, j \leq k$,

$$(1 + c')^{-1} \leq \frac{|z_i - z_j|}{|y_i - y_j|} \leq (1 + c'). \tag{1.1}$$

Near Extensions and Alignment of Data in $\mathbb{R}^n$*: Whitney extensions of near isometries, shortest paths, equidistribution, clustering and non-rigid alignment of data in Euclidean space*,
First Edition. Steven B. Damelin.
© 2024 John Wiley & Sons Ltd. Published 2024 by John Wiley & Sons Ltd.

(1) *Does there exist a c-distortion* $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ *which obeys* $\Phi(y_i) = z_i,\ 1 \le i \le k$*?*
(2) *Is it possible that* Φ *at the same time agrees with Euclidean motions as well?*
(3) *Can one say something about how* c, c', k, d *are related?*

Problem 2. *Let us be given a positive constant c small enough depending on d. Does there exist a positive constant* c' *small enough depending on c so that the following holds? Given two sets of* $k \ge 1$ *distinct labeled points in* $\mathbb{R}^d$, $\{y_1, ..., y_k\}$ *and* $\{z_1, ..., z_k\}$*. Suppose for every* $1 \le i, j \le k$*, (1.1) holds.*

(1) *Is it possible to find a Euclidean motion A which obeys* $A(y_i)$ *is close to* z_i *for every* $1 \le i \le k$*. Here close depends on c and the points* $\{y_1, ..., y_k\}$ *and is measured in the Euclidean norm.*
(2) *Can one say something about how* c, c', k, d *are related?*

Remark 1. *A central remark, at this juncture, is needed moving forward. Problem 1 and Problem 2 are fundamentally different in the sense that Problem 1 is a problem dealing with the existence of extensions. Problem 2 does not ask for an extension. It asks for a Euclidean motion only. This fact translates itself in many ways, for example in how the constants* c, c', k, d *relate to each other.*

In the case of isometry, Remark 1 is far less subtle: indeed, the following result is well-known, see for example [112].

Let $\{y_1, ..., y_k\}$ and $\{z_1, ..., z_k\}$ be two collections of $k \ge 1$ distinct points in $\mathbb{R}^d$. Suppose that the pairwise distances between the points are equal, that is, the two sets of points are isometric. That is

$$|z_i - z_j| = |y_i - y_j|,\ 1 \le i, j \le k.$$

Then, there exists a Euclidean motion, $A : \mathbb{R}^d \to \mathbb{R}^d$ with

$$A(y_i) = z_i,\ 1 \le i \le k.$$

1.3 Variant 2

In this section, we provide some perspective on Variant 2. We briefly discuss the interpolation problem in the sense of manifold learning in Chapter 4.

1.4 Visual Object Recognition and an Equivalence Problem in $\mathbb{R}^d$

Visual object recognition is the ability to perceive properties (such as shape, color and texture) of a visual object in $\mathbb{R}^d$ and to apply semantic attributes to it (such as identifying the visual object). This process includes the understanding of the visual object's use, previous experience with the visual object, and how it relates to the containing space $\mathbb{R}^d$. Regardless of the object's position or illumination, the ability to effectively identify an object, makes the object a "visual" object.

One significant aspect of visual object recognition is the ability to recognize a visual object across varying viewing conditions. These varying conditions include object orientation, lighting, object variability, for example, size, color, and other within-category differences to name just a few. Visual object recognition includes viewpoint-invariant, viewpoint-dependent and multiple view theories to name just a few examples. Visual information gained from an object is often divided into simple geometric components, then matched with the most similar visual object representation that is stored in its memory to provide the object's identification. See the following references and the many cited therein [1, 2, 4–6, 19, 39–42, 59–62, 66, 67, 69, 75, 82, 85–88, 91–93, 100, 101, 107, 111, 118, 120].

With this in mind, we define what we mean by an equivalence problem in $\mathbb{R}^d$. Imagine we are given two visual objects O and O' in $\mathbb{R}^d$. An equivalence and symmetry function $g : O \to O'$, when well defined, is an element of a group. See [98].

Some examples of vision maps are:

(a) Affine functions: a function $A : \mathbb{R}^d \to \mathbb{R}^d$ is an affine function if there exists a linear transformation $M : \mathbb{R}^d \to \mathbb{R}^d$ and $x_0 \in \mathbb{R}^d$ so that for every $x \in \mathbb{R}^d$, $A(x) = Mx + x_0$. Affine functions preserve area (volume) ratios. If M is invertible (i.e., A is then invertible affine), then A is either proper or improper. If M is not invertible, the function A is neither proper nor improper.
(b) Euclidean motions.
(c) Reflections: a reflection $A : \mathbb{R}^d \to \mathbb{R}^d$ is an isometry with a hyperplane as a set of fixed points.
(d) Similarity functions: a Euclidean motion plus a scaling. Similarity functions preserve length ratios.
(e) Projective motions, $(x, y) \to \left(\dfrac{ax + by + e}{a_1x + b_1y + e_1}, \dfrac{a_2x + b_2y + e_2}{a_3x + b_3y + e_3}\right)$ with

$$\det \begin{bmatrix} a & b & e \\ a_1 & b_1 & e_1 \\ a_2 & b_2 & e_2 \\ a_3 & b_3 & e_3 \end{bmatrix} = 1.$$

(f) Camera rotations, projective orthogonal transformation:

$$\begin{bmatrix} a_1 & b_1 & e_1 \\ a_2 & b_2 & e_2 \\ a_3 & b_3 & e_3. \end{bmatrix}$$

$\in SO(3)$.

(g) Motion tracking (video group). $(x, y, t) \to (x + at, y + bt, t)$.

- Here, in (e-g), a, b, e, a_i, b_i, e_i, $1 \leq i \leq 3$ are certain real constants. See for example [98].

1.5 Procrustes: The Rigid Alignment Problem

The recent advances in $d = 3$ data acquisition and the increasing interest in augmented and virtual reality have led to an explosion of volumetric data, as exemplified by point clouds. Point cloud data is prevalent in numerous applications, including robotics, autonomous driving, medical imaging, neuroscience, social science and computer graphics. In many of these applications, the captured point clouds correspond to noisy observations of an object/scene undergoing different deformations. One of the core challenges in these applications is to perform point cloud registration, which refers to finding a transformation that aligns or partially aligns the source and target point sets. At a high level, any point cloud registration algorithm must solve two problems: 1) finding accurate correspondences between the points in the source and target point clouds (implicitly or explicitly), and 2) modeling the deformation to match corresponding source and target points. The existing methods then propose different correspondence estimation algorithms and/or propose novel deformation modeling approaches.

The registration/deformation map (i.e., the transformation) could be rigid, or non-rigid. Most of the existing works in the literature have focused on rigid registration of point clouds, as it is a more prevalent problem in classic computer vision tasks like simultaneous localization and mapping (SLAM). The core innovations in these approaches are often with regards to finding the right correspondences between the points. For instance, the classic iterative closest point (ICP) algorithm relies on nearest-neighbor correspondences as measured via the Euclidean distance between points. This section deals with rigid matching. See [1]. The best way to understand the rigid alignment problem is via Procrustes alignment. The classical rigid Procrustes problem is to find a rigid motion that best aligns two

Figure 1.1 Orthogonal Procrustes fish (Rigid): From Simon Ensemble.

given point-sets in the least-squares sense. More precisely, given two sets of $k \geq 1$ distinct points in $\mathbb{R}^d$, say $\{y_1, ..., y_k\}$ and $\{z_1, ..., z_k\}$. The rigid Procrustes problem is the optimization problem

$$\min_{M \in O(d)} \sum_{i=1}^{k} ||M(y_i) - z_i + x_0||_2^2.$$

where $||.||_2$ is the l_2 norm. Here, we recall that $O(d)$ is the orthogonal group of $d \times d$ orthogonal matrices. The alignment is label-wise. Unlabeled problems are challenging, given it is often unclear which point to map to which. See for example our work in Chapter 21.

The rigid Procrustes optimization problem (See Figures 1.1–1.2), has a closed form solution obtained by applying a singular value decomposition (SVD) to a matrix created from the points $\{y_1, ..., y_k\}$ and $\{z_1, ..., z_k\}$. See [105]. For $d = 2, 3$, the rigid Procrustes problem and its generalizations for example to (a) unlabeled problems or/and (b) robustness to outliers, see [1, 59, 61, 62, 93] and the many references cited therein, is a well-known problem with diverse applications for example in computer vision, graphics, robotics [61, 62, 93, 101, 119, 120], chemistry [2], morphology [19], and many others. There are many interesting variants of the rigid Procrustes problem for other norms therein. See also Procrustes analysis with Wasserstein distances, Chapter 22. We do not pursue these directions in this monograph.

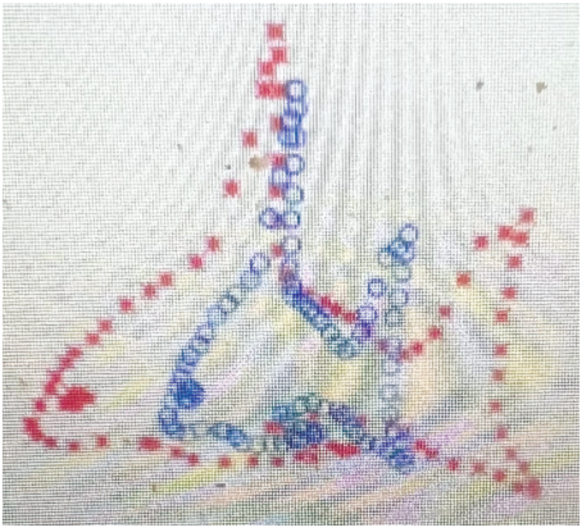

Figure 1.2 Orthogonal Procrustes fish (Rigid): From Simon Ensemble.

1.6 Non-rigid Alignment

In many real-world applications, the deformation between two sets of point cloud data is inherently non-rigid (See Figures 1.3–1.4). For instance, in medical imaging, the point cloud data could come from the surface of a tissue (e.g., liver) or several spiking neurons describing Alzheimer's disease (e.g, in the human brain), both, which can undergo non-rigid deformations. Such nonlinear deformations are generally modeled using two categories of approach, namely parametric and non-parametric approaches. In the parametric approaches, the deformation is characterized via a parametric function, e.g., parameters of an affine transformation or thin plate spline (TPS) parameters, and they are optimized to minimize the expected distance between corresponding source and target points. The non-parametric approaches, on the other hand, directly calculate the displacement (velocity) between source points and their corresponding target points. The existing approaches for non-parametric non-rigid point cloud registration vary in how to estimate the velocity of each point and how to regularize the velocity vector field for coherency and smoothness.

In this section, we rely mostly on two well-known papers: [91, 101]. Non-rigid shape matching often has different challenges to rigid matching even when the space of deformations is limited to, e.g., near isometries and one reason for this is that non-rigid shape matching methods often lead to challenging non-linear, non-convex optimization problems and hence challenging algorithms for robustness. One of the main reasons for this is that unlike in the case of a rigid deformation, non-rigid shape matchings are most frequently represented as pairings

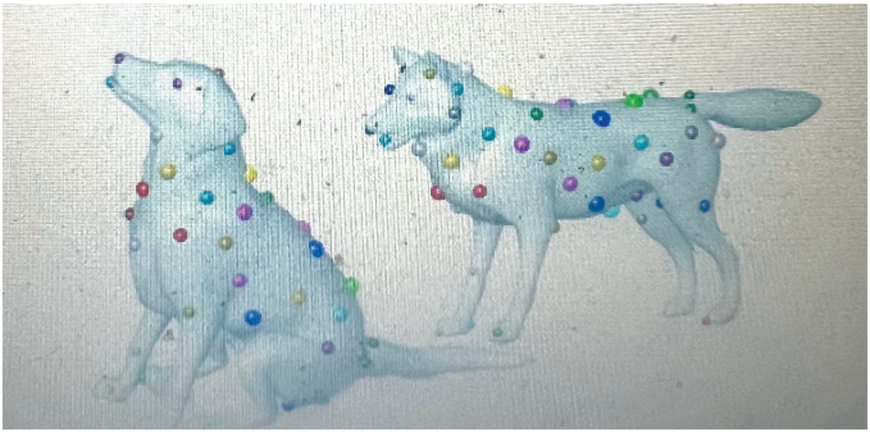

Figure 1.3 Near isometric correspondence shape matching found between a dog and a wolf: see reference [91].

Figure 1.4 Near isometric correspondences found between horses: see reference [101].

(correspondences) of points or regions on the two shapes being matched. Finding correspondences between a given discrete set of points on two different surface grids is an important problem in graphics, geometric processing, and computer vision. For example, applications include shape interpolation, surface completion, statistical shape modeling, symmetry analysis, shape matching, and deformable surface tracking. Intrinsic shapes for objects of the same class are often isometric, and are sometimes composed of big sections that are near isometric. The examples of the wolf, horse, and dog are typical. Exponential sizes of possible point correspondences, support constraints, and shape ambiguity, for example, present challenges to non-rigid matching even in the case of near isometric matching. Landmark matching, function space matching, and matching via Mobius transformations are just three of many interesting ideas known in the study of these difficulties. See [60, 91, 100, 101] and the many authors and papers referenced therein for more details.

Variants 1–2 represent a new idea in the study of near isometric alignment of data. We begin, thus, to build machinery to study Problems 1-2 in Section 2.1.