1 INTRODUCTION

Geometry is unavoidable.

One of the earliest branches of mathematics, geometry naturally attracted the attention of early artists, scientists, philosophers, and engineers seeking order in the world around them. It is hardly a leap of faith to find value in geometric reasoning: Complex geometric figures abound in nearly any environment. Organic and man-made figures alike are informed by their interaction with space, and nearly any task in architecture, art, mechanics, or physics begins with reasoning about shape.

Jumping forward several millennia, computational tools developed to engineer the Pyramids and reason about Platonic solids no longer suffice. With the advent of 3D scanning, additive manufacturing, advanced medical devices, and high-resolution imaging, modern algorithms must cope with complex curved and noisy shapes. Computational systems must gracefully store, manipulate, and find meaning in shapes composed of intertwined handles, singularities, and regions of varying curvature. Notions of bending and stretching are intertwined with *semantic* meaning; the folds of a brain's gray matter, for example, are not only geometric features but carry functional value derived from evolution and biological development. These challenges inspire advanced algorithms in computer vision and digital geometry processing that assemble multimodal, noisy, and conflicting signals about shape into a coherent model of the environment suitable for robotic guidance, computer-aided design (CAD), custom manufacturing, prosthetic design, and a host of other tasks.

Even more recently, the intuition we developed for low-dimensional shapes embedded in three-dimensions has proven valuable for high-dimensional problems involving abstract clouds of data points. Reasoning about distances, similarities, embedding, and structure makes perfect sense when discussing types of data we do not typically associate with shape, from corpora of text to flows of users through the links on a website. Some recent exploratory techniques are even designed to identify low-dimensional manifold structure in data for which a simple geometric explanation is not readily available.

Merging our understanding of these and other appearances of geometry in computation suggests the necessity and broad application of *geometric data processing*, considered in two senses:

1. $\underbrace{\text{Geometric data}}_{\text{Object}}\ \underbrace{\text{processing}}_{\text{Action}}$: Algorithms for processing geometric data, and
2. $\underbrace{\text{Geometric}}_{\text{Adjective}}\ \underbrace{\text{data processing}}_{\text{Action}}$: Data processing using geometric techniques.

With these related but distinct applications in mind, here we aim to widen the scope of "geometric data processing" from a specialized branch of statistics (see e.g., [12, 13]) to a broad field encapsulating the mathematical theory, algorithms, and computational applications of shape processing applied to abstract datasets and scans of physical objects alike. By combining theoretical underpinnings with algorithm development and detailed understanding of key applications across disciplines, we can develop geometric problem-solving methodologies with rigorous foundations and the potential to be used across the sciences and humanities.

In these notes, we attempt to merge insight from various corners of the computational world into a general and (at times) unified approach to "Shape Science."[1] Our focus is neither on algorithmic complexity nor on theorem-proving, but rather we demonstrate how ideas from differential and

1 With apologies to my friend and colleague Prof. Amir Vaxman, who continues to try to "make 'Shape Science' happen."

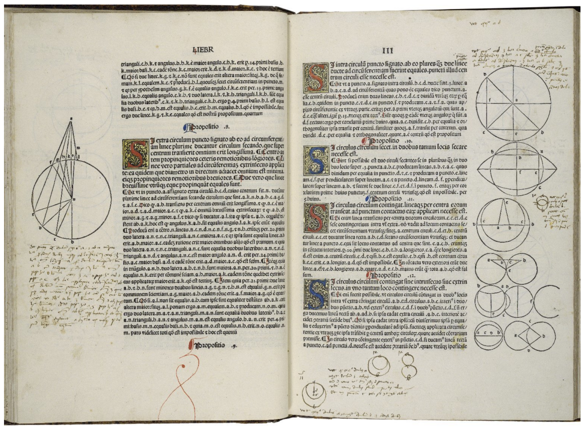

Figure 1: The first printed edition of Euclid's *Elements* (1482).
Image credit: Folger Shakespeare Library Digital Image Collection (INC E86)

discrete geometry form a powerful modeling toolbox. Even as computer science moves toward data-driven rather than prescriptive techniques, the appearance and reappearance of common geometric themes and techniques throughout past applications underscores the relevance of basic skills in this discipline.

1.1 THE MATHEMATICIAN'S TOOLKIT

Although engineers continue their formal education in calculus and algebra into college, typically their first and last encounter with geometry is in a single high school class focusing on techniques pioneered by Euclid circa 300 BC. Euclid's *Elements* no doubt establish fundamental ideas of proof and logic, but from a modern geometric perspective they are lacking. A quick glance at a page from the *Elements* (see Figure 1) reveals an issue: The shapes it considers are too simple! While a compass and straightedge may suffice for measuring and constructing polygons and circles, the reader is challenged to use the same equipment to extract equally accurate measurements from coffee mugs, door handles, fingers, and toes.

Centuries of effort have gone into broadening geometric language and theory, providing the formalism needed to measure and compare objects of varying form, dimensionality, roughness, and connectivity. Here, we briefly highlight just a few of the many branches of modern mathematics we will employ in developing machinery for geometric data processing:

DIFFERENTIAL GEOMETRY. Shortly after calculus revolutionized the way we think about functions, mathematicians began applying analogous techniques to analyzing shape. This reasonable application of derivatives and integrals led to the development of *differential geometry*, which understands shape through the lens of infinitesimal calculations. Differential geometry will be the primary tool we will use in the first part of our discussion, dedicated to local analysis of geometric structure.

The basic object of interest in differential geometry is the *manifold*, a locally k-dimensional object. For example, $k = 1$ defines a curve and $k = 2$ defines a surface. The earliest treatments of differential geometry required manifolds to be embedded in d-dimensional Euclidean space, where $d \geq k$, but later purely intrinsic versions were able to relax this assumption. By equipping manifolds with means of computing angles, distances, and areas, e.g. using the restriction of $\mathbb{R}^d$ to an embedded manifold, one can define notions of curvature, shortest paths between points (geodesic curves), tangency, differentiation of functions, and deformation.

Draw me!

Figure 2: Riemannian geometry is based on the observation that the length of a vector is determined by its base point.

The main difficulty initially is to define all these objects without dependence on a choice of coordinate system: Quantities in differential geometry "should" depend only on shape rather than on the choice of a parameterization. Hence, a large part of introductory differential geometry involves calculations to show that a given definition—perhaps for convenience given in terms of a mapping from k-dimensional space into the manifold—is independent of the choice of mapping. After these basic constructions, classical differential geometry derives elegant relationships between local and global geometry. For instance, the celebrated Gauss–Bonnet theorem shows that integrating local measures of curvature gives a topological invariant, that is, a value dependent only on the number of holes in a surface distinguishing e.g. donuts and coffee mugs from tennis balls. Other threads of differential geometry characterize local behavior of vector fields, describe what it means for a manifold to have bounded or constant curvature, and understand the effects of perturbing an object's shape on angles and areas, to name a few big ideas from this gargantuan branch of mathematics.

DIFFERENTIAL EQUATIONS. Differential equations specify relationships between derivatives of a function; for example, Newton's second law $F = ma$ relates acceleration—a second derivative in time—and force—typically a function of position, velocity, and time. Both ordinary (derivatives in one variable) and partial (derivatives in multiple variables) differential equations arise as natural tools for expressing and analyzing relationships between different quantities on manifolds and other pieces of geometry.

Ordinary differential equations (ODEs) are used to describe *parallel transport*, the procedure of dragging a vector along a manifold while maintaining tangency without spinning unnecessarily; parallel transport on the plane involves simply displacing the base point of a vector, but on a curved surface the vector must change to account for changing tangent planes from point to point. Behavior of physical partial differential equations (PDEs) like the heat equation qualitatively reflects the geometry of a shape; a metal plate, for instance, conducts heat differently than a metal sphere. Finally, *geometric flows* are PDEs that modify geometry over time; a famous example is mean curvature flow, which performs gradient descent on surface area to produce *minimal surfaces* like soap films.

RIEMANNIAN GEOMETRY. Arguably a branch of differential geometry, but one worth calling out independently for its immense contributions to the geometry world, Riemannian geometry is a model that divorces shape from topology, or connectivity. As a motivating example, consider a map of the world rolled flat onto a table. The map is sufficient to understand which land masses are connected to which: A path drawn in pencil along the map can be realized as a path on the surface of the earth. On the other hand, we know instinctively that the geometry of the map as a sheet of paper is meaningless: Distances between points on the paper measured in inches are not true distances along the surface of the earth.

Bernhard Riemann's model of geometry considers *Riemannian manifolds*, which couple topological spaces (e.g. the sheet of paper with a map drawn on it) with a means of computing lengths and angles on the true geometry (e.g. the longitude/latitude lines). The latter is encapsulated as an inner (dot) product function per point on the map, which can vary depending on where you are. As illustrated in Figure 2, if you draw a one-inch vector on the map based in South America, the length of that vector interpreted as a vector on the surface of the earth would be different from the length of the same one-inch vector based in Antarctica, which is usually more stretched out on a planar map of the earth.

Theorems in differential geometry take elegant form in Riemannian language, and a generalization to four-dimensional "pseudo-Riemannian manifolds" forms the basis for Einstein's theory of general relativity. In the computational context, we will be less concerned with physics and more with how Riemannian formalism broadens the scope of ideas of curvature and vector fields to higher-dimensional manifolds relevant to data analysis. We will minimize our use of daunting mathematical notation appearing in the Riemannian world largely due to Albert Einstein; application-oriented computer scientists are understandably scared away thanks to expressions like the following for the (extremely practical) Laplace–Beltrami operator:

$$\Delta f = \frac{1}{\sqrt{|g|}}\partial_i\left(\sqrt{|g|}g^{ij}\partial_j f\right).$$

The geometric intuition for operators like Δ is often quite simple, however, despite the unfamiliar language.

METRIC GEOMETRY. Manifolds are relatively "strong" geometric structures, in the sense that the requirements on a space that lead it to be an embedded or Riemannian manifold are relatively strong. In contrast, some models of geometry are weaker, allowing us to apply intuition about shape to a much broader class of problems.

As one particularly general case, *metric geometry* involves the geometry of metric spaces, or spaces that admit metric functions. A metric $d(x, y)$ is distinguished by just a few properties:

$$\begin{array}{ll} d(x,y) \geq 0 & \text{(nonnegativity)} \\ d(x,y) = 0 \iff x = y & \text{(Leibniz's Law)} \\ d(x,y) = d(y,x) & \text{(symmetry)} \\ d(x,y) \leq d(x,y) + d(y,z) & \text{(triangle inequality).} \end{array}$$

These four properties axiomatize what it means to be a "distance." For example, the path from one point to another has the same length as the path in the reverse direction (symmetry), and the distance directly from one point to another is never more than the length of a path that stops at some intermediate location (triangle inequality).

Many spaces admit metric structure even if they are not manifolds. A critical example in the computational world is a *graph* or network, described as a collection of nodes connected by edges. On a graph, a distance metric is obtained by computing shortest-paths along edges. Even though objects like graphs may not admit derivatives and tangent spaces, geometric reasoning can go a long way toward designing algorithms that process metric-structured data efficiently and with theoretical justification.

OPTIMAL TRANSPORT. While the luckiest data scientist may find that a cloud of data points truly admits differentiable structure, most datasets are noisy and incomplete. To address this issue, recent techniques have explored the feasibility of extending geometric data processing to the case where features are known with uncertainty. One promising theoretical tool in this regime is *optimal transport*, which lifts distances between points to distances between probability distributions. This way, even if we do not know the exact location of a feature point on a surface but just a fuzzy idea of where it might be, we can still attempt to carry out computations downstream such as computing shortest paths. Transport has also proven valuable in machine learning, providing a geometrically-motivated alternative to constructions in information theory used to measure similarity of probability distributions, suitable for parameter estimation and inference.

1.2 COMPUTATIONAL THEMES

Existence and uniqueness proofs hardly suffice for practical purposes. As computer scientists and engineers working with geometric data, we must take relevant models and ideas from the

Draw me!

Figure 3: Many possible ways to represent a shape.

theoretical literature and make them work numerically. This task presents new challenges at the interface of abstraction and application. Here we mention a few of the many questions that must be answered when designing a modern geometric algorithm.

WHAT IS SHAPE? A central design decision in any geometric system is the choice of *representation*. There are many options, illustrated in Figure 3:

- The simplest geometric representation is a *point cloud*, or collection of points $\{x_1, \ldots, x_k\}$ embedded in Euclidean space $\mathbb{R}^n$. Point clouds are the easiest geometric representations to obtain and store but are unstructured; they do not, for example, indicate which points compose a local neighborhood around a center point.

- *Meshes*, composed of points connected into networks of basic elements like triangles and tetrahedra, give connectivity information not present in a point cloud. The complexity of a mesh depends on the dimensionality of the data, however, and it is not clear how to assemble a mesh from a point cloud if it is not given.

- Computer aided design (CAD) benefits from geometric representations of surfaces and volumes as *smooth parameterized functions*, e.g. polynomials or rational functions. This representation allows for exact computation of differential quantities like curvature, but the designer is restricted to a smaller space of possible shapes. It is an open problem to stably assemble a minimal collection of parameterized patches approximating a point cloud or mesh.

- *Implicit* methods represent a manifold as the set $\{x \in \mathbb{R}^n : f(x) = 0\}$ for some function $f(x) : \mathbb{R}^n \to \mathbb{R}$. The function $f(x)$ can be stored in any number of ways, from samples on a grid to a sum of basis functions like Gaussians. Unlike meshed representations, implicit representations can transition elegantly between different topologies, making them a key representation in the fluid simulation world. On the other hand, storing values of $f(x)$ far away from the zero level set is wasteful, making it difficult to achieve a high-resolution expression of geometry.

- *Graphs* equipped with edge lengths are a first example of geometry that is not embedded in $\mathbb{R}^n$ but can be suitable computationally for understanding the geometry of road networks and other thin structures.

- *Metric spaces* can be discretized in the most general as *symmetric matrices* of pairwise distances between elements of the metric space. This representation is generic but takes a quadratic amount of space and does not admit calculus and other fine-grained computation.

The decision of which geometric representation make sense for a given problem has huge bearing on which operations are and are not possible or efficient, and a poor choice of representation is difficult to reverse in late stages of an engineering project.

Although many applications dictate the required representation, in others areas this truly is a degree of freedom. An example is machine learning from shape, e.g. learning to map from a 3D model to a label or segmentation. There exists little consensus on the most effective shape representation for these problems, and the choice of mesh vs. point cloud vs. implicit vs. other options has strong bearing on performance, availability of training data, and ease of training/evaluation.

DISCRETE OR DISCRETIZED? Quantities in differential geometry such as curvature are defined in terms of derivatives along a manifold. But, consider the triangle mesh in Figure REF. How do

Draw me!

Figure 4: Zooming out: From geometry "in the small" to geometry "in the large."

we speak about its geometry? Is it piecewise flat with infinite curvature at the vertices? Or, do we view it as a sampling of a smooth surface?

Considerable discussion in geometric computing revolves around the theme of *discretized* vs. *discrete* quantities. The push-pull between these two approaches distinguishes different techniques applied to the same problem or data:

- *Discretized* models apply numerical analysis to approximate differential quantities on a sampled piece of geometry. Mathematical discussion about discretized geometry revolves around the *convergence* of an approximation as the piece of geometry is sampled more and more densely in the computational representation, as in Figure REF. Convergence theorems ensure that algorithmic approximations are representative of a smooth model, but they often do not provide as much insight on behavior in the finite-data regime. For example, a discretization of curvature on the mesh in Figure REF might fit a smooth quadratic patch to a neighborhood of each vertex and use the curvature of that patch as an approximation to the curvature of the surface at the vertex.

- *Discrete* models deal with geometry as it exists before taking the limit of refinement. These often redefine notions from differential geometry to be compatible with finitely-sampled models even if they are sampled coarsely. Discrete measures of curvature on the mesh in Figure REF might be defined directly in terms of interior angles and edge lengths of the triangles rather than requiring fitting smooth surfaces to the data locally. A focus of discrete differential geometry (DDG) is on *structure preservation*, the idea that combinatorial analogs of theorems about smooth differential geometry sometimes can be proved exactly in the discrete case, before taking the limit of refinement. Certain "no free lunch" theorems, however, show that you cannot "have it all" CITE: Preserving one property exactly may force another to be violated under certain computational models CITE.

We will illustrate differences between discrete and discretized differential geometry starting with our treatment of curves in Chapter 3.

HOW FAST? Algorithms for computer graphics must contend with a "magic number:" 29.97 frames per second. This is the frame rate of video for color televisions and monitors; algorithms that generate output slower than this bound will visibly lag. Software in medical imaging has a different trade-off, since it might be acceptable to forego efficiency for more accurate diagnosis; but at the same time, if a medical image analysis algorithm takes too long, a doctor's appointment may end, making it difficult to bring the patient back for a second scan if the data is too noisy to process.

The trade-off between accuracy and efficiency is well-known to computer scientists. It is not hard to approach asymptotic complexity problems in the geometric domain; simple meshes of two-dimensional shapes reach thousands to millions of vertices and simplices. Implementation and modeling options for controlling runtime include the density of the geometric representation—several adaptive algorithms attempt to resample geometric domains to use fewer points while capturing the same features—and complexity of the model. In the end, geometric algorithms range from sparse matrix inversion, to discrete geometry computations like meshing and finding Voronoi cells, to solving large-scale optimization problems with one variable per sample point; each incurs a different cost in terms of number of iteration and complexity of a single step.

1.3 OUR APPROACH

As our discussion revolves around shape, it is only fitting to organize this course using a geometric structure: *scale*. Our approach is illustrated pictorially in Figure 4.

After some preliminaries, we begin our discussion at the micro scale, asking what we can learn about a shape by holding a magnifying glass up to a single point. We will rediscover a theme well-known to early differential geometers, that an ant walking along a surface can sense certain geometric features simply by remembering where it has been and how far it has walked. This *intrinsic* perspective can be paired with additional *extrinsic* data, which characterizes how a piece of geometry interacts with the space around it; that is, we give our ant the ability to look up from the ground and observe the nearby horizon. Most of our discussion here will be in the two-dimensional case of surfaces embedded in 3D space, the most common appearance of geometry in computer vision and graphics.

Pulling our camera out slightly, we consider relationships between two or more points on a piece of geometry through use of distances. Spaces in which we can compute distances between points are known as *metric spaces*, which encapsulate not only curved surfaces but also many other weaker geometric objects that may not even be embedded in Euclidean space. This notion not only provides an alternative abstraction of what it means to be a shape, but also leads to a fundamental inverse problem: Which geometric spaces can be embedded into one another? For example, clearly a one-dimensional line can be embedded many ways into two-dimensional space, but the two-dimensional plane cannot be squeezed onto a line without inadvertently gluing points together. The theory and practice of embedding will suggest applications in texture mapping for computer graphics as well as principal component analysis (PCA) and multi-dimensional scaling (MDS) in statistics.

We take a slight detour to consider an intersection of algebra and geometry provided by Lie groups and representation theory. In Lie groups and Lie algebras, local geometric structure not only defines shape but also symmetries and relationships. For example, the group of planar rotations $\mathrm{SO}(2)$ is parameterized by the set of angles $0^\circ - 360^\circ$, which inherits the geometry of the unit circle S^1. As with many topics covered here, we will only highlight a few interesting and potentially useful computational applications of Lie groups and algebras, for which a thorough treatment would require an entire textbook. Even so, we will derive some useful models and algorithms appearing in robotics and microscopy, for which careful treatment and understanding of Lie groups leads to theoretically well-justified algorithms for cleaning up data with extremely low signal-to-noise ratios.

Next, we zoom out to see a whole shape instead of just a few isolated points. In particular, we develop a notion of *calculus* on a curved geometric figure, including vector fields, rates of change, integration by parts, and other techniques familiar from college mathematics. While we typically think of the basic operators from calculus—e.g. "div, grad, curl, and all that" [16]—as differential in nature, inverting these operators by solving partial differential equations (PDEs) gives us a *global* view of geometry inspired by physical intuition; we find that the geometry of a shape determines how it conducts heat and vibrates. Indeed, the reverse direction of realizing a shape from its physical properties suggests a famous problem in modern geometry with bearing on shape retrieval algorithms: Can you hear the shape of a drum? [11] While we will see (*spoiler alert!*) that there exist differently-shaped drums that sound the same, we will also use vibration modes and related quantities computable from a single powerful operator—the *Laplacian*—to derive algorithms in shape comparison and semisupervised machine learning.

Continuing to use ideas from calculus to characterize global geometry, we will study the mathematics of vector fields and flows, a basic tool in differential geometry needed to understand signals as they change and flow along a geometric domain. We will motivate some ideas from Cartan's exterior calculus, used to generalize basic vector field constructions to higher dimensions. This formalism is readily adapted to triangulated surfaces and other discrete structures through discrete exterior calculus (DEC), which translates exterior calculus operators to sparse matrices that remarkably preserve topological structures from the smooth case, applied with success to 3D surface editing and meshing problems.

We wrap up our discussion of calculus on manifolds with two applications. First, we apply the differential operators we have developed to the shape itself, modeling *geometric flows* that can smooth out geometries and eliminate or form singularities. We also take a brief detour to explore computational applications of *optimal transport*, a theory linking probability to geometry that recently has gained traction as a modeling tool for posing problems in both shape matching and high-dimensional probabilistic reasoning.

Discussion of optimal transport also transitions our perspective to include more than one shape at a time. As an additional, we consider the task of *segmenting* a geometric object into meaningful connected pieces, with application to image/shape segmentation as well as unsupervised clustering of data points embedded in a geometric space. Afterward, our main focus in geometric comparison is on *correspondence*, in which we aim to extract a smooth, low-distortion, and semantically-meaningful mapping from one geometric space into another. Basic machinery for this task will apply to registration of point clouds from a 3D scanner, to alignment of medical images and other models to transfer labels or deformations, to abstract transfer learning algorithms designed to cope with distributional differences between training and test data. We will consider three instances of correspondence: rigid alignment, in which shapes are rotated and translated into the same coordinate system; nonrigid alignment, in which the shapes can also be bent and/or stretched to align; and finally intrinsic correspondence, in which a mapping $\phi : M \to N$ from space M to a space N is extracted without explicitly warping one onto the other.

We conclude with a panoramic view of geometry, jointly considering all members of large shape collections or databases. Our discussion begins with a decidedly theoretical—and somewhat philosophical—question: Can we put a geometry on the set of shapes? That is, does the "space of spaces" [19] itself have geometric structure? Although this infinite-dimensional question sounds abstract, it leads to means of measuring distances between shapes, helping sort out the progression of brain surfaces through the course of a neurodegenerative disease, as well as means of computing a "shortest path" from one shape to another, providing smooth in-betweening animations between poses of a character designed by an artist. Our remaining discussion is more concrete, examining how a collection of shapes can be put into consistent correspondence, how machine learning tools can be applied to unstructured shape data, and finally how some recent techniques exploring how to endow a computer with the creative ability to generate *new* shapes given representative samples from a collection.

1.4 WHAT IS NOT COVERED

In a one- to two-semester course, it is impossible to shed light on every detail of theoretical or applied geometry. These notes are by no means comprehensive but rather are intended as an introduction to geometric reasoning in a computational context, including informal highlighting of unexplored frontiers in this active branch of research.

Here we quickly note the limits of our discussion as a disclaimer to readers seeking a particular approach or discussion of a particular topic:

- While we attempt to provide self-contained discussion of relevant mathematics, this text should *not* be considered a comprehensive introduction to the theory of differential geometry; we refer the reader to classic texts such as CITE for more careful discussion. In an effort to provide intuition about a broad mathematical toolset, many theorems from differential geometry will be quoted and motivated informally rather than proved in generality.

- Similarly, while we do explore constructions of discrete notions of curvature, calculus on manifolds, and related topics, we take a pragmatic approach to shape analysis that includes both *discrete* and *discretized* geometry in parallel. For detailed discussion of discrete differential geometry (DDG), a mathematical theory that attempts to define a self-contained theory of differential geometry on simplicial complexes, see CITE.

- We consider applications of geometric data in 3D geometry processing, computer graphics, machine learning, medical imaging, and other domains. While the core challenges faced in these disciplines are often geometric, equally important are the many applied developments that make geometric tools compatible with the particularities of these disciplines. For detailed discussion of specific applications, see e.g. CITE.

It goes without saying that any text is largely a reflection of the biases, interests, and knowledge of the author. The author here offers his sincere apologies and a Venti-sized beverage to any colleagues who feel their work has been omitted or discussed irresponsibly herein.