

Mathematical Foundations in Visualization

Ingrid Hotz^{*}, Roxana Bujack[†], Christoph Garth[‡], Bei Wang[§]

Abstract

Mathematical concepts and tools have shaped the field of visualization in fundamental ways and played a key role in the development of a large variety of visualization techniques. In this chapter, we sample the visualization literature to provide a taxonomy of the usage of mathematics in visualization, and to identify a fundamental set of mathematics that should be taught to students as part of an introduction to contemporary visualization research. Within the scope of this chapter, we are unable to provide a full review of all mathematical foundations of visualization; rather, we identify a number of concepts that are useful in visualization, explain their significance, and provide references for further reading.

We assume the reader has basic knowledge of linear algebra [90], multivariate calculus [89], statistics, combinatorics, and stochastics [39]. Other topics not covered in this chapter, such as image analysis [88], computer graphics [86], signal processing [41], computational geometry [18], geometric modeling, mesh generation, computer aided geometric design [35, 106], and numerics [76], can be found in well-established textbooks.

1 Data and basic terminology

You can have data without information, but you cannot have information without data.

Daniel Keys Moran, programmer and science fiction writer

Data are at the center of every visualization task and every step of the visualization pipeline, see Fig. 1.

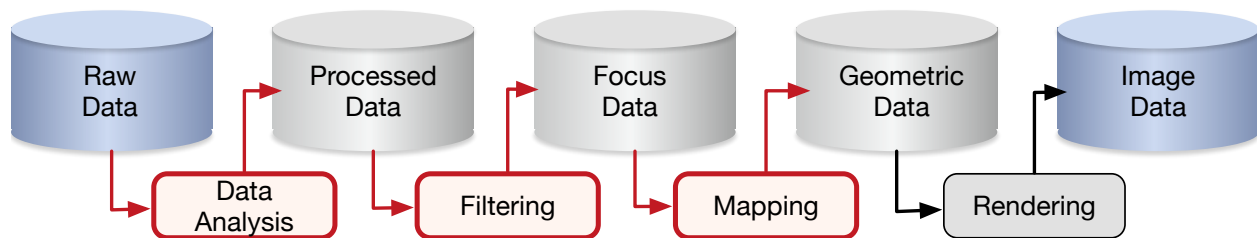


Figure 1: A visualization pipeline. All steps in the pipeline involve the use of mathematical concepts and tools. We cover various aspects of data analysis, filtering, and mapping.

The input to the visualization pipeline, the raw data, can be any collection of information in any form. In this chapter, we define a data set as a triplet $\mathcal{D} = (\mathcal{S}, \mathcal{A}, f)$ consisting of a set of structured items \mathcal{S} , a set of attributes \mathcal{A} , and a function that assigns attributes to the items. \mathcal{S} consists of a set of items, continuous or discrete, together with a structure (such as a metric for a continuous domain or neighborhood relations for networks), see Fig. 2 for an example.

^{*}ingrid.hotz@liu.se, Linköping University, Sweden

[†]bujack@lanl.gov, Los Alamos National Laboratory, USA

[‡]garth@cs.uni-kl.de, Technische Universität Kaiserslautern, Germany

[§]beiwang@sci.utah.edu, University of Utah, USA

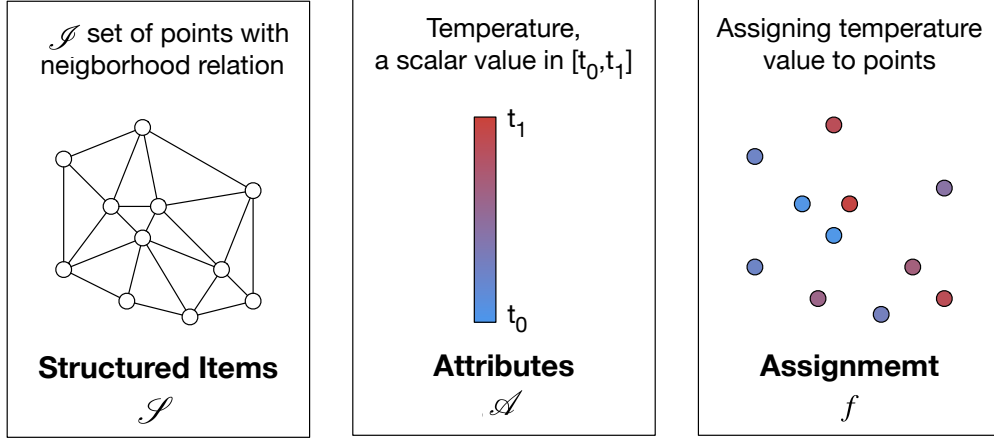


Figure 2: An example of a data set: \mathcal{S} consists of a set of points with a neighborhood relation. Attributes in \mathcal{A} are elements of the interval $[t_0, t_1]$. f assigns temperature values to the points.

The tools used for the analysis and visualization of data sets depend on the nature of \mathcal{S} and \mathcal{A} . The most important distinctions are continuous vs. discrete structures, and quantitative vs. categorical attributes, see Table 1. In this section, we emphasize continuous structures and quantitative attributes. A more detailed classification of data sets concerning types, structures, and organizations can be found in Munzner [68]. An introduction to data representations from a scientific visualization perspective can be found in Telea [92].

The structure \mathcal{S}

The structure \mathcal{S} can vary from discrete points to continuous domains. In general, \mathcal{S} consists of a set of items and some relation between the items. We describe two of the most frequently used structures in more detail.

GRAPHS, NETWORKS AND TREES. *Graphs* or *networks* are structures that are frequently used for non-spatial, relational data representations. The terms *graph* and *network* are sometimes used interchangeably. Mathematically, a graph G is a pair (V, E) consisting of a set of items V , called vertices or nodes, and a set of relationships between these items expressed as a set of edges $E \subseteq V \times V$. Edges can be directed or undirected. For directed graphs, (v, w) , and $(w, v) \in E$ represent different relations. If the edges are assigned a numeric attribute, the graph is weighted.

A possible representation of a finite graph is an adjacency matrix, which is a square matrix of size $|V| \times |V|$. For a simple graph, the adjacency matrix is a $(0,1)$ -matrix with zeros on its diagonal and ones for each edge. If the graph is undirected, the matrix is symmetric. Typically, graphs are displayed using a set of points for the vertices, which are joined by lines for the edges. A general introduction to graphs and networks can be found in [56].

When analyzing graphs, characteristics as cycles, planarity, sparseness, and hierarchical representations are of interest.

Structures \mathcal{S}	Attributes \mathcal{A}
continuous domains equipped with metrics	ordered, ordinal, quantitative
meshes, simplicial complexes	scalars, vectors, tensors
graphs, networks, trees	categorical

Table 1: Examples for possible structures \mathcal{S} and attribute spaces \mathcal{A} .

CONTINUOUS DOMAINS. A continuous *domain* D is a subset of \mathbb{R}^n equipped with a *metric*. A metric supports measurements and determines distances in the domain. A common metric is the Euclidean distance. Other metrics include Manhattan distances and polar distances. More generally, when the domain is a *parameterized manifold*, the choice of a metric has an impact on many calculations such as derivatives, see Section 2.

A continuous domain can be represented by a finite set of discrete samples associated with an interpolation scheme. In this case, \mathcal{S} consists of a set of points $\{p_i \in D \mid i = 1, \dots, k\}$, equipped with a neighborhood structure; e.g., the points are organized as a *regular grid* (associated with piecewise multilinear interpolation) or a simplicial complex (corresponding to piecewise linear interpolation).

The attribute space \mathcal{A}

An attribute is a specific property assigned to data items that arise from measurement, observation or computation. Attributes can be continuous and quantitative, e.g., temperature; discrete and ordered, e.g. the number of people in a class; as well as categorical, e.g., various types of tree species. The set of possible attributes span the *attribute space*.

The most common *continuous quantitative attributes* can be subsumed under the term *tensor*. A *tensor of order r* is defined as a multi-linear mapping acting on r copies of a n -dim vector space V over \mathbb{R} into the space of real numbers,

$$\mathbf{T} : \underbrace{V \times \dots \times V}_r \rightarrow \mathbb{R}. \quad (1)$$

Sometimes *rank*, *degree* and *order* are used interchangeably. A tensor of order 0 corresponds to a scalar $\alpha \in \mathbb{R}$ and a tensor of order 1 is a vector $\mathbf{v} \in V$.

$$\begin{aligned} \alpha : \mathbb{R} &\rightarrow \mathbb{R}, & \alpha(x) &= \alpha x & \text{0th-order tensor or } & \textit{scalar}, \text{ e.g., temperature;} \\ \mathbf{w} : V &\rightarrow \mathbb{R}, & \mathbf{w}(\mathbf{v}) &= \mathbf{w} \cdot \mathbf{v} & \text{1}^{\text{st}}\text{-order tensor or } & \textit{vector}, \text{ e.g., velocity;} \\ \mathbf{T} : V \times V &\rightarrow \mathbb{R}, & \mathbf{T}(\mathbf{v}, \mathbf{v}') &= \mathbf{v} \cdot \mathbf{T} \cdot \mathbf{v}' & \text{2}^{\text{nd}}\text{-order } & \textit{tensor}, \text{ e.g., strain tensor.} \end{aligned}$$

Tensors of higher order, especially 3rd- and 4th-order tensors, can also be found in a few visualization applications. In the visualization literature, the term tensor often refers to 2nd-order tensors. With respect to a specific basis $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ of the vector space V , a tensor is fully specified by its action on the basis elements resulting in the typical component representations. For a vector, this is $w = (w_1, \dots, w_n)^T$ and for a 2nd-order tensor, this is a matrix

$$\mathbf{T} = \begin{pmatrix} t_{1,1} & \cdots & t_{1,n} \\ \vdots & \ddots & \vdots \\ t_{n,1} & \cdots & t_{n,n} \end{pmatrix}.$$

For a basic introduction to the use of tensors in visualization, we refer to the state-of-the-art report by Kratz et al. [59].

ENRICHED ATTRIBUTE SPACE \mathcal{A}^* . In-depth data analysis often requires some modifications of the attribute space. The most common examples are *filtering*, e.g. removing noise, or *enrichment* of the original attributes by derived quantities, e.g. the field gradient or local histograms. Other modifications are changes of the representation or parameterization of the attribute space to emphasize data symmetries useful for feature or pattern definitions; see also Section 4. Examples include scaling, rotation in attribute space, and expressing a 2nd-order tensor by its eigenvalues and eigenvectors.

Fields as example data sets

Field data are very common in scientific applications where they express physical quantities defined over continuous domains, for instance, temperatures in a room, or wind velocities in the atmosphere. Such data are often the results of numerical simulations or measurements from experiments. A *field* is defined

as a mapping from a continuous *domain* $D \subseteq \mathbb{R}^n$ into an *attribute space* $\mathcal{A} \subseteq \mathbb{R}^m$ (similar notions include *range* and *co-domain*), given as

$$f : \mathbb{R}^n \supseteq D \rightarrow \mathcal{A} \subseteq \mathbb{R}^m. \quad (2)$$

Typically, the domain can be considered in a spatiotemporal context, for example, $D = D_s \times I_t \subseteq \mathbb{R}^4$, where $D_s \subseteq \mathbb{R}^3$ is the spatial domain and $I_t \subseteq \mathbb{R}$ is a time interval. Depending on the attribute space, we distinguish a *scalar field* $S : D \rightarrow \mathcal{A} \subseteq \mathbb{R}$, a *vector field* $V : D \rightarrow \mathcal{A} \subseteq \mathbb{R}^2$, a *tensor fields* T , and more generally, a combination of such fields, resulting in a *multifield* with an attribute space spanned by the individual fields.

ENSEMBLES OF FIELDS. Fields are often associated with a set of parameters, which typically play a different role than the domain dimensions. Parameters are often used to create collections of data sets, referred to as *ensembles* [96].

$$\{f_1, \dots, f_k\} : D \times \{P_1, \dots, P_k\} \rightarrow \mathcal{A} \subseteq \mathbb{R}^m, \quad (3)$$

where each P_i (for $i = 1 \dots k$) is a parameter tuple. An example of an ensemble is the data set generated from a computer simulation with different initial conditions (described by different parameters). Each $f_i : D \times P_i \rightarrow \mathcal{A}$ is an *ensemble member* or a *realization*. Ensemble members often have internal correlations or follow certain distributions, making them especially hard to analyze. Ensemble data arise in many applications and is an important theme in visualization research [46].

2 Differential structures

Science is a differential equation. Religion is a boundary condition.

Alan Turing, mathematician and computer scientist

Whereas real data and computations are mostly based on discrete domains and attributes, many of the concepts for their analysis are founded on continuous settings. The machinery of differential arithmetics and differential structures provide powerful analysis tools. *Differential operators* [15, 87] play a crucial part in visualization. They allow the definition and categorization of many features, including *extrema*, *ridges*, *valleys*, *saddles* and *vortices*. *Differential equations*, for example, are the basis for the definition of streamlines, a fundamental method in flow visualization, see Fig. 3(a).

Finally, *differential geometry* provides mathematical tools to characterize curves and surfaces and plays an important role in visualization, see Fig. 3(b). In this chapter, we summarize the most fundamental concepts of discrete structures that are frequently encountered in visualization research.

Differential operators

DIFFERENTIAL OPERATORS IN EUCLIDEAN SPACES. *Differential operators* [15, 87] map functions (e.g. fields) to their derivatives and thus allow us to study the rates at which continuous attributes change. They can be applied to scalar, vector, and tensor fields. They give rise to definitions of features, such as *extrema*, *ridges*, *valleys*, *saddles*, and *normals* of isosurfaces. We describe differential operators for scalar fields $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and vector fields $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$. The explicit expression of the operators depends on the inherent metric of the space; here, we assume the Euclidean metric. We often use the operator

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \dots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \quad (4)$$

to simplify the notations. The *gradient* of a scalar field

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \dots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}, \quad (5)$$

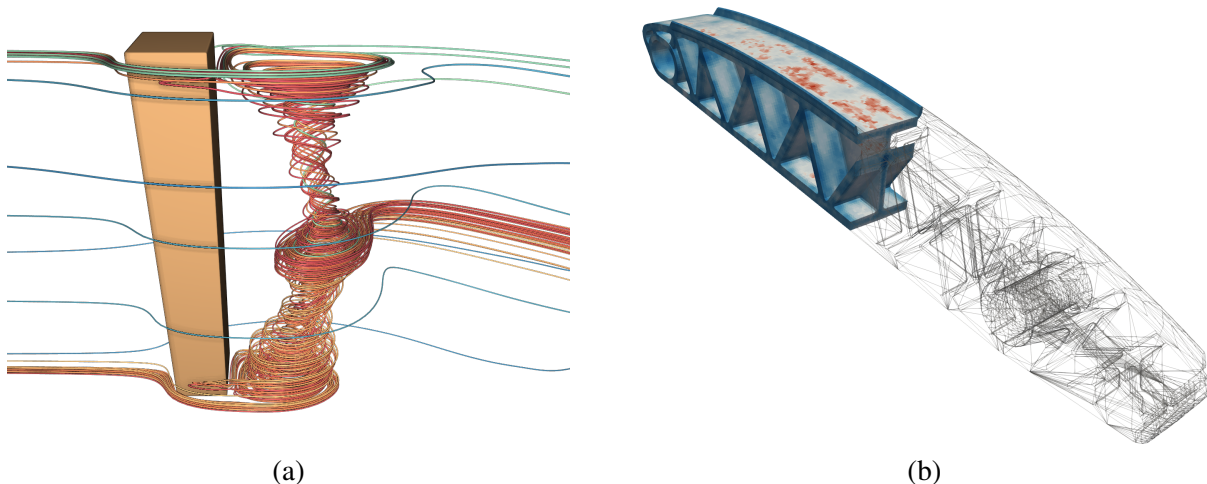


Figure 3: An interplay between discrete data and continuous concepts. (a) Numerically computed streamlines of the flow behind a cylinder approximate the solutions of an ordinary differential equation. (b) A discrete mesh approximates the shape of a mechanical part, where a continuous color map highlights the extremal values of the load of the material.

is a vector that indicates the direction of the steepest ascent. Locations where gradient vanishes ($\nabla f = 0$) are associated with *critical points* of the scalar field, such as maxima, minima, and saddles, see also Section 7. *Hessian matrices* consisting of 2^{nd} -order partial derivatives are used to classify the critical points,

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial f}{\partial x_1 \partial x_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}. \quad (6)$$

The eigenvalues of the Hessian H can be interpreted as the principal curvatures, and the eigenvectors as principal directions; therefore H is often used to define ridges and valley lines in scalar fields. For example, a *topographic ridge* is defined as the set of points where the slope is minimal on the scalar field restricted to a contour line. This means that one eigenvector of H is aligned with the elevation gradient [72].

The *Jacobian* $J \in \mathbb{R}^{n \times n}$ is a matrix that generalizes the concept of a gradient for a vector field v ,

$$J = \nabla v = \begin{pmatrix} \frac{\partial v_1}{\partial x_1} & \cdots & \frac{\partial v_n}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial v_1}{\partial x_n} & \cdots & \frac{\partial v_n}{\partial x_n} \end{pmatrix}. \quad (7)$$

The eigenvalues of the Jacobian can be used to categorize the types of 1^{st} -order *critical points* in vector fields, i.e., positive for sources, negative for sinks, differently signed for saddles, and complex for center points, see Fig. 4.

Other important differential operators are the *Laplace operator* $\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$, the *divergence* $\text{div } v = \nabla \cdot v$, and the *curl* $\text{curl } v = \nabla \times v$ of a vector field. In an infinitesimal neighborhood, the divergence is a measure of how much the flow converges toward or repels from a point, and the curl indicates of how much the flow swirls or rotates.

DIFFERENTIAL OPERATORS FOR FIELD APPROXIMATIONS. Differential operators also play an important role in the approximation of fields as they represent the components in the *Taylor expansion*. A scalar field in the vicinity of a point $P \in \mathbb{R}^n$ can be approximated as $f(P+x) = f(P) + \nabla f(P) \cdot x + \frac{1}{2} x^T H(P) x + O(\|x\|^3)$. For vector fields, the linear approximation is given as $v(P+x) = v(P) + J(P) \cdot x + O(\|x\|^2)$.

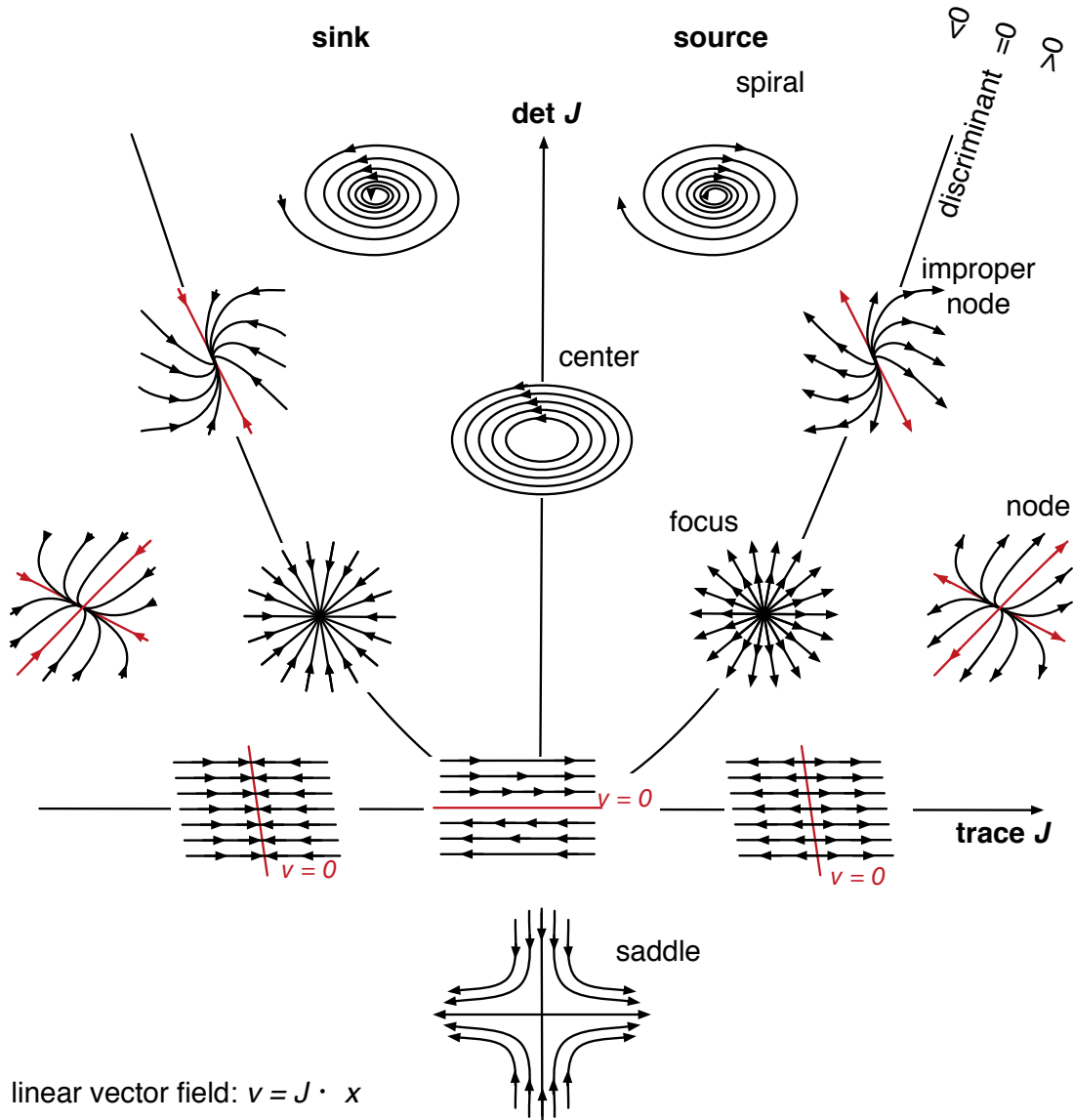


Figure 4: The Jacobian J can be used to classify the local behavior of a vector field in the vicinity of a critical point. Locally, the field can be approximated up to 1st order via the Taylor expansion as $v(x) = v_0 + J \cdot x$. If $v_0 = 0$, the point x is critical. The critical point can be classified based on the determinant and the trace of the Jacobian. The sign of the discriminant $\Delta = \text{tr}^2(J) - 4\det(J)$ separates the area of real and complex eigenvalues of the Jacobian. Complex eigenvalues are associated with swirling motions.

DIFFERENTIAL OPERATORS IN NON-EUCLIDEAN SPACES. For non-Euclidean spaces, differential operators are more complex. Consider, for example, spherical coordinates: the divergence of a vector $(v_r, v_\theta, v_\varphi)$ (where r is the radius, θ is the polar angle, and φ is the azimuthal angle) is then given as

$$\text{div } v = \frac{1}{r^2} \frac{\partial (r^2 v_r)}{\partial r} + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (v_\theta \sin \theta) + \frac{1}{r \sin \theta} \frac{\partial v_\varphi}{\partial \varphi}. \quad (8)$$

The differential operators for cylinder and spherical coordinates can be found in most textbooks.

Differential equations

A *differential equation* [1, 80] is a mathematical equation that relates a function with its derivatives. Differential equations are categorized into ordinary differential equations (containing one independent variable), and partial differential equations (involving two or more independent variables).

One of the most common examples of an ordinary differential equation in visualization is given through the relation of a vector field and its trajectories (that are everywhere tangential to the field), see Fig. 3(a). A flow can be represented either as a time dependent vector field $\mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, $(x, t) \mapsto v(x, t)$ or through its *flow map*,

$$\mathbb{R} \times \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad t \times t_0 \times x_0 \mapsto F_{t_0}^t(x_0), \quad (9)$$

$$\text{with } F_{t_0}^{t_0}(x_0) = x_0, \text{ and } F_{t_1}^{t_2}(F_{t_0}^{t_1}(x_0)) = F_{t_0}^{t_2}(x_0).$$

The flow map describes how a flow parcel at (x_0, t_0) moves to $F_{t_0}^{t_1}(x_0)$ in the time interval $[t_0, t_1]$. The two representations of the vector field are related through the initial value problem [14],

$$\dot{F}_{t_0}^t(x_0) = v(x(t), t), \quad F_{t_0}^{t_0}(x_0) = x_0, \quad (10)$$

where \dot{F} refers to the temporal derivative of F , and inversely through integration,

$$x_0 + \int_{t_0}^t v(x(t), t) dt = F_{t_0}^t(x_0). \quad (11)$$

Partial differential equations are more complex than ordinary differential equations, and, depending on the initial and boundary conditions [36], may not have a unique solution or a solution at all. As a popular example, we can look at the heat equation,

$$\frac{du(x, t)}{dt} - \alpha \nabla^2 u(x, t) = 0, \quad (12)$$

Where $\alpha \in \mathbb{R}$ is called the *thermal diffusivity*. The solution to the above heat equation is a Gaussian. It describes the physical problem of heat transfer or diffusion and is used in various visualization applications, for instance, in diffusion-based smoothing, or to define a continuous scale space.

Even if solutions of differential equations exist, for visualization applications, it is rarely possible to derive them analytically, but only numerically [10, 66], due to the reliance on empirical data for coefficients, initial conditions, and boundary conditions. The most popular solvers for ordinary differential equations are the Euler and Runge-Kutta methods. For partial differential equations, the families of finite element methods (FEM), finite volume schemes, and finite differences methods are frequently used, depending on the choice of discretization.

Differential geometry

We review elements from differential geometry [61] that are most relevant to visualization, including parametrized curves and surfaces, lengths, areas, and curvature. Some of these concepts can be generalized from three-dimensional to higher-dimensional spaces dealing with general manifolds, which are topics in Riemannian Geometry [2].

PARAMETRIC CURVES. In differential geometry, curves are defined in a parametrized form, and their geometric properties, including arc length, curvature, and torsion, are expressed using integrals and derivatives. A *parametric curve*

$$\gamma: [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}^d \quad (13)$$

is a vector-valued function defined over a non-empty interval. Curves can be distinguished depending on how often they are differentiable. In the continuous case, we will assume the curve to be sufficiently smooth.

The fundamental theorem of differential geometry of curves guarantees that up to transformations of the Euclidean space (rotations, reflections, and translations), a three-dimensional curve can be uniquely defined by its *velocity*, *curvature*, and *torsion*. These three concepts describe changes of the *Frenet-Serret*

frame, which is a local coordinate system that moves with the curve. A Frenet-Serret frame is spanned by the unit *tangent* vector $T(t)$, *normal* vector $N(t)$, and *binormal* vector $B(t)$, which are defined via derivatives of the curve $\gamma(t)$ with respect to the parameter $t \in [a, b]$,

$$\begin{aligned} T(t) &= \frac{\gamma'(t)}{\|\gamma'(t)\|}, \\ N(t) &= \frac{\gamma''(t) - (\gamma''(t) \cdot T(t))T(t)}{\|\gamma''(t) - (\gamma''(t) \cdot T(t))T(t)\|}, \\ B(t) &= T(t) \times N(t). \end{aligned}$$

Consequently, commonly used curve descriptors include *velocity* $v(t) = \|\gamma'(t)\|$, *curvature* $\kappa(t) = \|T'(t) \cdot N(t)\|/\|T(t)\|$, and *torsion* $\tau(t) = \|N'(t) \cdot B(t)\|/\|T(t)\|$. Other useful measures are the *arclength* $l(t) = \int_a^t \|\gamma'(s)\| ds$ and the *acceleration* $a(t) = \gamma''(t)$.

PARAMETRIC SURFACES. Similar to curves, surfaces can be parametrized, see Fig. 5. A parametric surface

$$S : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n \quad (14)$$

is a vector-valued function of a non-empty area. We assume the surface to be sufficiently smooth.

The *tangent plane* of a surface at a point $S(p) \in \mathbb{R}^n$ with $p \in \mathbb{R}^2$ is the union of all tangent vectors of all curves through $S(p)$. The plane is spanned by the two partial derivatives $S_u(p) = \partial S / \partial u$ and $S_v(p) = \partial S / \partial v$. The *surface normal*, perpendicular to the tangent plane, is given by the cross product of the partial derivatives,

$$N(p) = \frac{S_u(p) \times S_v(p)}{\|S_u(p) \times S_v(p)\|}. \quad (15)$$

MEASUREMENTS ON SURFACES. The calculation of the length of a curve on a surface or the surface area can be easily formulated using the *first fundamental form* $\mathbf{I}(p) : \mathbb{R}^2 \rightarrow \mathbb{R}^{2 \times 2}$. $\mathbf{I}(p)$ defines a natural local metric induced by the Euclidean metric in \mathbb{R}^n . For notational simplicity, we omit the dependence of the location $p \in \mathbb{R}^2$. Its components g_{uv} are defined as the scalar product of the tangent vectors $S_u \cdot S_v$. In matrix form, the first fundamental form is given as,

$$\mathbf{I} = \begin{pmatrix} g_{uu} & g_{uv} \\ g_{vu} & g_{vv} \end{pmatrix} = \begin{pmatrix} E & F \\ F & G \end{pmatrix}. \quad (16)$$

Using the first fundamental form, a line element ds on the surface is expressed as $ds^2 = Edu^2 + 2Fdu dv + Gdv^2$ and an area element as $dA = \|S_u \times S_v\| du dv = \sqrt{EG - F^2} du dv$. The arclength of a curve on the surface results from integrating the line element $l = \int_a^b ds$, and the area of a surface patch results from integrating the area element.

SURFACE CURVATURE. Many different curvature measures are available. Loosely speaking, curvature is a concept that measures the amount by which a surface deviates from a plane or the variation of the surface normal. Central to the concept of curvature is the *Gauss map*, which maps the surface normals to the unit sphere $N : S \rightarrow S^2$. The differential of the Gauss map in a certain direction is a measurement of curvature in that direction. Mathematically, the curvature is summarized in the *second fundamental form*, denoted as \mathbf{II} . In matrix form, it is given as,

$$\mathbf{II} = \begin{pmatrix} S_{uu} \cdot N & S_{uv} \cdot N \\ S_{vu} \cdot N & S_{vv} \cdot N \end{pmatrix} = \begin{pmatrix} e & f \\ f & g \end{pmatrix}, \quad (17)$$

where S_{uv}, S_{uu}, S_{vv} are the respective second derivatives of the the surface parametrization. The *shape operator* expresses the curvature in *local coordinates*,

$$S = \frac{1}{EG - F^2} \begin{pmatrix} eG - fF & fG - gF \\ fE - eF & gE - fF \end{pmatrix}. \quad (18)$$

Its eigenvalues (k_1 and k_2) are called the *principal curvatures* at a given point; and its eigenvectors are called the *principal directions*. The *Gaussian curvature* K is equal to the product of the principal

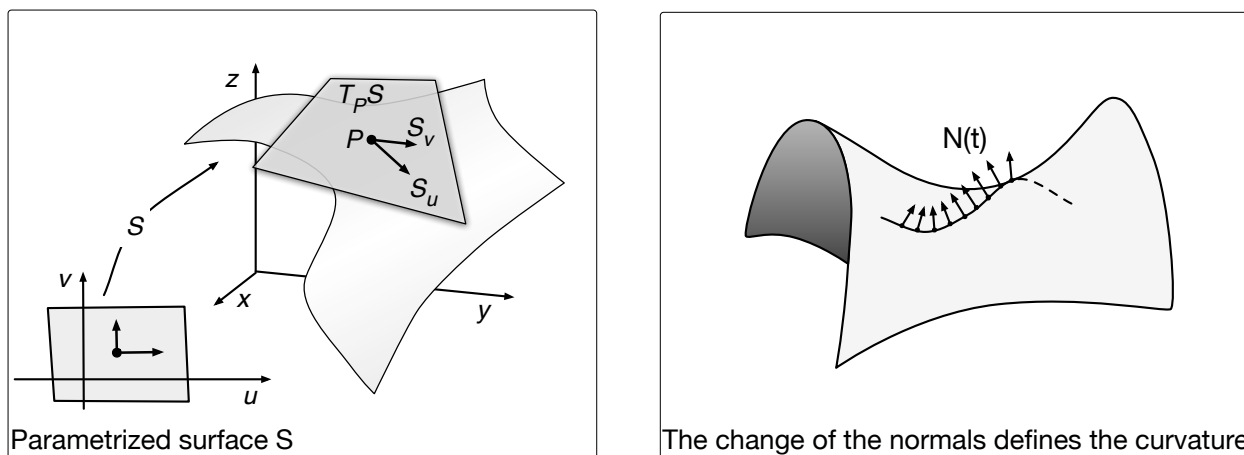


Figure 5: Left: parametrized surface. Right: The changes of the normals in a certain direction define the normal curvature of the surface.

curvatures. It can also be calculated as the ratio of the determinants of the second and first fundamental forms. The *mean curvature* H is defined as the average of the principal curvatures:

$$K = k_1 \cdot k_2 = \frac{eg - f^2}{EG - F^2}, \quad H = \frac{1}{2}(k_1 + k_2) = \frac{1}{2} \frac{eG + gE - 2fF}{EG - F^2}.$$

Points on the surface can be categorized as *elliptic* ($K > 0$), *parabolic* ($K = 0, H \neq 0$), *hyperbolic* ($K < 0$), and *flat* ($K = H = 0$) using the Gaussian and mean curvatures.

The curvature κ of a surface curve γ can be decomposed into its *normal curvature* k_n normal to the surface and its *geodesic curvature* k_g , which measures the deviation of a curve from being a geodesic $\kappa^2 = k_n^2 + k_g^2$. The extrema of the normal curvature over all curves through a point correspond to the principal curvatures k_1 and k_2 of the surface. A curve where the geodesic curvature is equal to zero is called a *geodesic*, which is a generalization of a straight line on arbitrary surfaces, as the straightest and locally shortest curve.

Manifolds

Roughly speaking, an n -manifold M embedded in \mathbb{R}^m is a space that is locally similar to Euclidean space \mathbb{R}^n . Formally, each point p of the manifold M has an open neighborhood $U_p \subset M$ that is homeomorphic to an open subset V of the Euclidean space described by a chart or local frame $\varphi : U_p \subset M \rightarrow V_p \subset \mathbb{R}^n$. The entire manifold can be described by a collection of compatible charts, which together form an atlas.

A well-known example is a sphere, which is a 2-manifold embedded in \mathbb{R}^3 , defined by the condition $x^2 + y^2 + z^2 = R$ (R being the radius). There are many ways to define charts on the sphere. It is also possible to cover the whole sphere excluding one point with a chart, which requires at least two charts to complete the atlas. Covering a sphere with one chart, however, is not possible.

Similar to surfaces, one can define a tangent space $T_p M$ attached to every point in M . $T_p M$ has the same dimension as the manifold. The tangent space defines a local basis on the manifold and plays an important role since many fields (e.g. vector fields) live in the tangent space of the domain, see Fig. 6.

3 Sampled Data and Discrete Methods

The world is continuous, but the mind is discrete.

David Bryant Mumford, mathematician

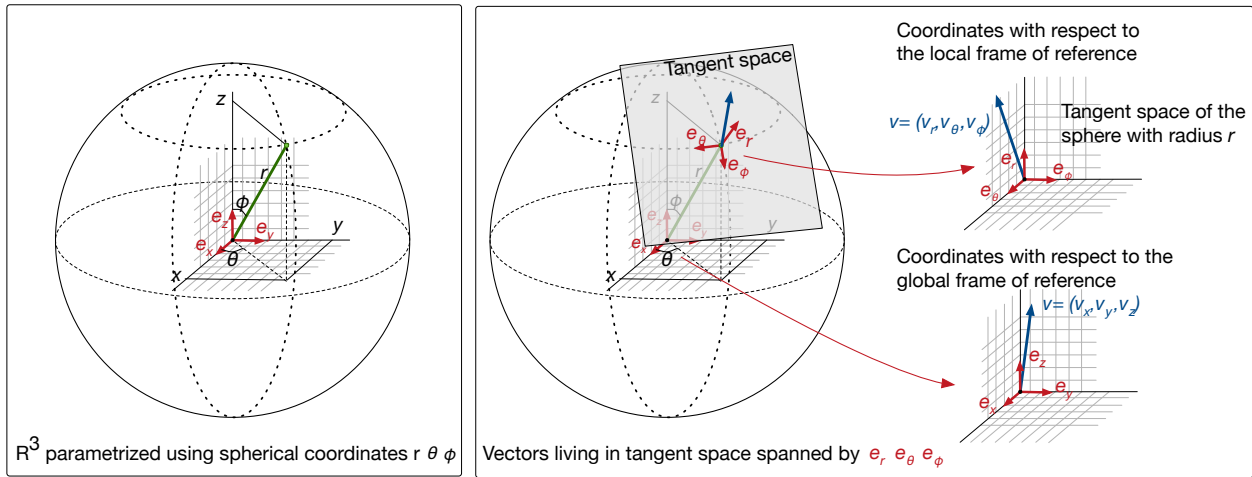


Figure 6: Vector field defined on a sphere given in spherical coordinates. Left: a parametrization of the sphere with spherical coordinates, Right: the vector field can be expressed in a local reference frame, which depends on the spherical coordinates.

Fields are defined over continuous domains in theory; however, they are described at discretely sampled locations in practice. Typical analysis and visualization methods rely on a *reconstruction* of the continuous fields. Two different approaches are commonly used to deal with this issue. First, the discrete data is interpolated to fill the entire domain. Second, the analysis techniques are transferred to the discrete setting.

Data representation

Sampled data come in many different forms and representations depending on their origin. For measurement data, one often deals with *unstructured point clouds* resulting from practical constraints, e.g., possible placements for sensors. Data coming from simulations are mostly based on grid structures, ranging from *uniform grids* to *unstructured* and *hybrid grids*. Therefore, the attributes are assigned to either the grid vertices, the grid cells, or distinguished points inside the cells, e.g., *Gauss* or *integration points* coming from finite element simulations, see Fig. 7. An overview of common data representations can be found in [92].

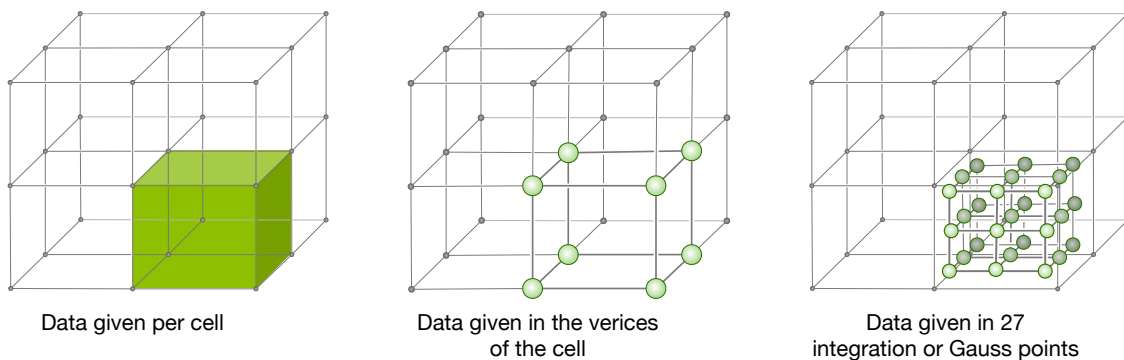


Figure 7: Data can be assigned to a regular cubic grid in many different ways.

A grid is built from a set of vertices V and neighborhood relations, defining edges, faces and cells. The neighborhood relations can be given explicitly for unstructured grids or implicitly encoded in an index structure. An example is a quad mesh where the vertices are identified by three indices $V = \{v_{ijk} \mid 1 \leq i, j, l \leq n\}$ and edges $E = \{(v_{ijk}, v_{i+1jk}), (v_{ijk}, v_{ij+1k}), (v_{ijk}, v_{ijk+1}) \mid \forall v_{ijk} \in V\}$. The most common 2-dimensional cells are triangles and rectangles; 3-dimensional cells include quads, tetrahedra, and prisms.

Simplicial complexes

Simplicial complexes are data structures that are particularly useful for combinatorial algorithms (see Section 7). They can be considered as a formal generalization of triangulations to higher dimensions. A k -simplex is defined as the convex hull of $k + 1$ affinely independent points $p_i \in \mathbb{R}^k$; the convex hull of any nonempty subset of the $k + 1$ points is a *face* of the simplex. 0-, 1-, 2- and 3-simplices are vertices, edges, triangles, and tetrahedra, respectively.

A *simplicial complex* K is a set of simplices such that every face of a simplex from K is also in K , and the intersections of two simplices in K is either empty or a face of both simplices, see Fig. 8. A more detailed discussion can be found in [67, 22]. A simplicial complex is a type of *cell complex* in which the cells are simplices. There are several different ways to formalize and instantiate the notion of a cell complex, including CW complex, Δ -complex, cube complex, polytopal complex, etc.; see Hatcher [47] for an introduction.

Neighborhood graphs

Neighborhood graphs impose combinatorial structures on point clouds that capture certain notion of proximity. Such structures give rise to the use of grid-based analysis methods but are also of interest for clustering algorithms and many discrete theories. The most fundamental neighborhood structure is the Delaunay triangulation of a point cloud. Given a finite set of points $P = \{p_i\}_{i=1}^m \subset \mathbb{R}^n$, the *Voronoi diagram* is defined as a decomposition of the domain in regions V_i assigned to each point $p_i \in P$. V_i contains all points in \mathbb{R}^n that are at least as close to p_i as to any other point in P . The dual structure of the Voronoi diagram in the plane is the *Delaunay triangulation* and in three dimension the *Delaunay tetrahedralization*. The Delaunay triangulation maximizes the minimum angle in a triangulation and gives rise to a reasonably nice triangulation. The concept extends to higher dimensions, but its computation becomes very costly. Many other neighborhood graphs have been studied with respect to geometric properties and robustness. Examples include the Gabriel graph [38] and the k -nearest neighbors graph. A more detailed discussion about such graphs can be found in textbooks on computational geometry [18]. Neighborhood graphs in the context of high-dimensional and sparse data in visualization applications are also discussed in [17]. There is a large body of work related to meshing that is also relevant in this context [106].

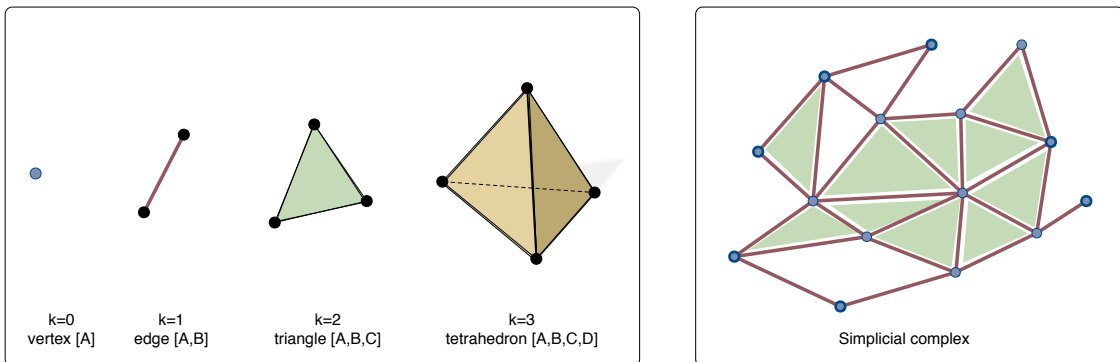


Figure 8: Left: 0-, 1, 2 and 3-simplex, respectively. Right: a simplicial complex embedded in \mathbb{R}^2 .

Reconstruction and interpolation

The goal of a *reconstruction* is to recover an approximate version of a continuous function from a sampled data set. A reconstruction that matches the values in the sampled points exactly is called *interpolation*.

Given a set of points (vertices or nodes) $P = \{p_i\}_{i=1}^m$ with $p_i \neq p_j$ for $i \neq j$ and a set of associated values $\{f_i \in \mathbb{R}\}_{i=1}^m$, a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called *interpolating function* for the set of points if it fulfills the interpolation condition $f(p_i) = f_i$, for $1 \leq i \leq m$.

Infinitely many possibilities are available to interpolate a set of points. The choice of a specific interpolation is often guided by simplicity and efficiency. It is important to be aware that different interpolation schemes may have significant impact on the computation and visualization results. The most common interpolation methods for gridded data are piecewise linear, bilinear, and trilinear interpolations. For scattered data, one typically constructs a grid or uses radial basis functions [6].

Discrete theories

Discrete theories typically inherit structural properties from the smooth setting and come with theoretical understandings about the preservation of relevant invariants. In general, they satisfy a subset of properties from the smooth setting, resulting in a large diversity of discrete theories [97]. For example, in the discrete setting, a geodesic defined as a locally shortest connection is not equivalent to the straightest connection, as in the continuous setting [73].

In visualization, the most important examples arise from combinatorial differential topology and geometry. For instance, discrete exterior calculus provides discrete differential operators [19]; and discrete differential geometry introduces concepts for curvatures and geodesics [20]. A very useful and popular discrete theory is discrete Morse theory [37], which forms the base of many current algorithms for the extraction of the Morse-Smale complex, see also Section 7.

4 Symmetries, Invariances, and Features

Symmetry is a vast subject, significant in art and nature. Mathematics lies at its root, and it would be hard to find a better one on which to demonstrate the working of the mathematical intellect.

Hermann Weyl, mathematician and theoretical physicist [98]

Symmetries, invariances, and conserved quantities are closely related concepts that play an important role in many mathematical and physical theories, for instance, Noether’s theorem links symmetries of physical spaces with conservation properties [84]. Invariants are properties of an object (a system or a data set) that remain unchanged when certain transformations (such as rotations or permutations) are applied to the object. In visualization, invariants play a central role for feature definition and pattern recognition. For example, the number of legs of a 3-dimensional animal model is invariant with respect to changes due to animal movement or shape morphing. Another example is the Galilean invariance for flow features, e.g., vorticity does not change under certain changes of the reference frame even though the flow components change [71]. There are also topological invariants which characterize spaces with respect to smooth deformations [47]. A formal analysis of the symmetries that arise from group actions, with a strong emphasis on the geometry, Lie groups, and Lie algebra, can be found in textbooks dealing with representation theory and invariant theory [42].

Features, traits, and properties

According to the Cambridge Dictionary, a feature is “a typical quality or an important part of something”. In the visualization literature, the term feature is not well-defined and oftentimes an overloaded concept. Features often represent structures in a data set that are meaningful within some domain-specific context. They can be used as the basis for abstract visualization. Here we define a feature $F(\mathcal{D}) \subset \mathcal{I}$ of a data set \mathcal{D} as a subset of data items having a specific property; see Section 1. For field data, features are

typically defined as certain subsets in the spatial domain. Typical features of a scalar field $s : D \rightarrow \mathbb{R}$ are iso-surfaces $s^{-1}(a)$ (for $a \in \mathbb{R}$), and the set of critical points of s .

In many cases, features can be locally defined by *traits* $T \subset \mathcal{A}^*$, subsets of the enriched attribute space \mathcal{A}^* containing the data attributes and possibly derived quantities. Specifically, given a field $f : D \rightarrow \mathcal{A}^*$ that maps a domain D into an enriched attribute space \mathcal{A}^* , a *trait-induced feature* is defined to be $F_T(\mathcal{D}) = f^{-1}(T) = \{x \in \mathcal{D} \mid f(x) \in T\}$, for some $T \subset \mathcal{A}^*$ [53]. A point trait $T = \{p\} \in \mathcal{A}^* = \mathbb{R}$ gives rise to a trait-induced feature known as an *iso-surface*. A point trait is also referred to as a *feature descriptor*. If \mathcal{A}^* encodes the derivatives of f , then the set of critical points is a trait-induced feature given by all points where the derivative of the scalar function is equal to 0. A line trait is a line in $\mathcal{A}^* = \mathbb{R}^2$ spanned by the scalar values and its derivatives. It is desirable for a descriptor to be invariant with respect to changes (e.g., rotations and scalings) to the data representation.

Other types of features based on structures of the data, such as cycles in a graph, may not be described by traits naturally. Such features are referred to as *structure-induced features*. In general, features can be defined by any combination of attribute and structural constraints.

Transformations, symmetries, and invariances

Invariants are directly linked to transformations T describing an inherent *symmetry* of the system. A *transformation* is a function that maps a set X to itself, i.e. $T : X \rightarrow X$. In the context of visualization, a transformation concerning the structure \mathcal{S} is called the *inner transformation*; a transformation of the attribute space \mathcal{A}^* is called the *outer transformation*. A transformation can be both an inner and an outer transformation. The notion of invariance and transformation can also be extended to changes in the model used to create the visualization, or the image itself [58].

When talking about invariants, we are interested not only in one specific transformation but also in certain classes of transformation described as transformation groups [42]. A transformation group acting on a set X is defined as a group G with neutral element e and an action

$$T : G \times X \rightarrow X,$$

where each group element $g \in G$ defines a transformation T_g as $T_g(x) \equiv T(g, x)$ with the following properties: for all $x \in X$ and all $g, h \in G$, $T_e(x) = x$ and $T_g(T_h(x)) = T_{gh}(x)$.

A *symmetry group* is a group that conserves a certain structure, property or feature. It gives a unique relation between symmetries and invariants. Formally, let $T : \mathcal{D} \rightarrow \mathcal{D}$ be a transformation (short for $T : \mathcal{S} \times \mathcal{A}^* \rightarrow \mathcal{S} \times \mathcal{A}^*$), and $F(\mathcal{D})$ be a feature of a data set \mathcal{D} . Then we say that T is a *symmetry* of \mathcal{D} if $F(\mathcal{D})$ commutes with the transformation T

$$T(F(\mathcal{D})) = F(T(\mathcal{D})).$$

Typical transformations for field data are rotations in 3-dimensional Euclidean space that form the group SO_3 acting on \mathbb{R}^3 . An application is the definition of invariant moments as descriptors of flow patterns [7]. An example that plays an important role in flow visualization is the Galilean transformation, which transforms coordinates between two reference frames that differ only by constant relative motion [57]. Domain-specific invariants like shear stress or anisotropy also play a central role in tensor field visualization [60]. An example of discrete data is the permutation group $Sym(M)$ whose elements are permutations of a set M .

5 Cluster Analysis

The Milky Way is nothing else but a mass of innumerable stars planted together in clusters.

Galileo Galilei, astronomer, physicist and engineer

A frequently employed approach in visualization and exploratory analysis is cluster analysis or clustering, i.e., to assign a set of objects to groups in a manner such that objects in the same group are more similar

to each other in some manner than to those in other groups. In other words, data are decomposed into a set of classes that in some sense reflect the distribution of the data.

To achieve this general goal, a very large variety of algorithms have been presented for specific problems or data modalities [51, 33]; they differ significantly in how they define and identify clusters. Clustering results are typically subject to various parameters, and it is often necessary to modify (e.g., transform) input data and choose parameters to obtain a result with desired properties. We describe four clustering techniques that are frequently applied in data analysis and visualization and illustrate how they have been used to address various visualization problems.

***k*-MEANS CLUSTERING.** Given a set of data (x_1, \dots, x_n) where each x_i is a d -dimensional real vector, *k*-means clustering (also called *Lloyd's algorithm*) seeks to partition the data into $k \leq n$ disjoint sets $C = \{C_1, \dots, C_k\}$ (with a fixed k) such that the variance within each cluster is minimized, i.e., to find

$$\arg \min_C \sum_{i=1}^k \sum_{x \in C_i} \|x - m_i\|^2$$

where m_i is the mean of data in C_i . The result depends centrally on the chosen metric, for which the Euclidean norm is often selected. Algorithmically, C_i can be found iteratively in a manner similar to computing a centroidal Voronoi tessellation [21]: given an initial set of cluster centers m_i , assign to each cluster the data points that are closer to m_i than to all other cluster centers. Compute a new set of means as cluster centers from the assigned points, and repeat the process until convergence. Initially, the data centers can either be chosen randomly or according to heuristics [12].

k-means clustering was used in visualization, for example, by Woodring and Shen [102], who employed it to automatically generate transfer functions for volume rendering temporal data. They achieved this by identifying clusters of data points that behave similarly over time. *k*-means clustering is relatively easy to understand and utilize. However, a major drawback of this approach is that the number of classes or clusters k must be specified a priori.

SPECTRAL CLUSTERING. Clustering is not directly applied on the data, but rather on the *similarity matrix* S (where $S_{ij} = \|x_i - x_j\|$) that contains pairwise distances between individual data items. Clustering is then performed on the eigenvectors of S . Intuitively, S can be viewed as describing a mass-spring system. Masses coupled through tight springs will largely move together relative to the equilibrium of such a system, and thus eigenvectors of small eigenvalues of S can be seen to form a suitable partition of the data.

As with clustering in general, many incarnations of this basic idea have been given. The *normalized cuts* technique is a non-parametric clustering approach often used in image segmentation [85]. For visualization purposes, it was utilized by Ip et al. to explore feature segmentation of three-dimensional intensity fields [50], and by Brun et al. to visualize white matter fiber traces in DT-MRI data [5].

DENSITY-BASED CLUSTERING. The DBSCAN (density-based spatial clustering of applications with noise) algorithm is a widely used general-purpose clustering scheme [32, 83]. It considers the density of data points in their embedding space and subdivides them into three types. A point x_i is a *core points* if at least m points lie within a distance of δ from x_i ; these points are called *directly reachable* from x_i . Both m and δ are parameters. An arbitrary point x_j is *reachable* from x_i if there is a path x_i, x_k, \dots, x_j such that each point in the path is directly reachable from its predecessor. Points that are not reachable from any core point are called *outliers*. Clusters are formed by core points and the points that are reachable from them. (There may be multiple core points in a cluster.) Due to the non-symmetric reachability relations, DBSCAN uses the notion of density-connectedness for a pair x_i and x_j . That is, points x_i and x_j are connected if there is a third point x_l from which both x_i and x_j are reachable.

DBSCAN is relatively easy to implement and has good runtime properties, but many variants of the basic technique exist that differ in various details [91, 83]. Wu et al. used DBSCAN to provide level-of-detail in visualization and exploration of academic career path [103].

MEAN SHIFT. A *mean shift* procedure is a variant of density-based clustering; it is applied to identify the maxima (or *modes*) of a density function from discrete samples. Fixing a kernel function $K(x_i - x)$

(typically flat or Gaussian) and a point x in the embedding space, the weighted mean in a window around x is

$$m(x) = \frac{\sum_{x_i} K(x_i - x)x_i}{\sum_{x_i} K(x_i - x)}.$$

The mean shift $m(x) - x$ is then minimized by setting $x \leftarrow m(x)$ and iterating until convergence. Data points x_j are grouped into clusters according to the mode to which the mean shift converges if initialized with x_j . This process yields a general-purpose clustering technique that does not incorporate assumptions about the data and relies on a single parameter, the kernel bandwidth. In visualization, a good example of the usefulness of this algorithm is given by Böttger et al. [3], who use mean-shift clustering to achieve edge bundling in brain functional connectivity graphs.

6 Statistics for Visualization

If the statistics are boring, you've got the wrong numbers.

Edward R. Tufte, statistician [93]

Statistics deals with the collection, description, analysis and interpretation of (data) *populations*. *Descriptive statistics* are used to summarize population data. *Moments*, also called *summary statistics*, are a statistical notion to describe the shape of a function (distribution). Mathematically, the n -th *central moment* of a real-valued continuous function $f(x)$ of a real variable is given by

$$\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx,$$

where c is the mean of $f(x)$. The first moment corresponds to the mean, and a usual assumption considers $c = 0$. These moments give rise to the usual statistical descriptors of a distribution such as variance ($n = 2$), skewness ($n = 3$), and kurtosis ($n = 4$). Potter et al. provide guidance on the visualization of functions via their summary statistics [75]. For multiple variables, the concept of moments can be generalized to *mixed moments*. Applications in visualization include pattern matching for feature extraction [8].

A frequent problem in comparative visualization is comparing distributions. Here, the *covariance* of two distributions

$$\text{cov}(f, g) = E[f - E[f]] E[g - E[g]]$$

signifies their joint variability. In the multivariate case, covariance can be generalized to the *covariance matrix*. Covariance matrices have been frequently used in visualization, for example in glyph-based [74] or feature-based visualization [101].

Furthermore, *correlation* of functions may be used for comparison. In the broadest sense, correlation is any statistical association between data populations; in practice, correlation is usually used to indicate a linear relationship between functions. An commonly used concept is the *Pearson's correlation coefficient*,

$$\rho_{f,g} = \text{corr}(f, g) = \frac{\text{cov}(f, g)}{\sigma_f \sigma_g},$$

where σ_f and σ_g refer to the standard deviation of f and g , respectively. $\rho_{f,g} \in (0, 1]$ if f and g are positively correlated; $\rho_{f,g} \in [-1, 0)$ if f and g are negatively correlated; $\rho_{f,g} = 0$ if f and g have no linear correlation. Finding correlations among data is one of the most essential tasks in many scientific problems, and visualization can be very helpful during such a process [13, 43].

Order statistics, on the other hand, characterizes a population in terms of ordering and allows us to make statistical statements about the distribution of its values. For example, the q -percentile ($0 \leq q \leq 100$) denotes the value below which q percent of the samples are located. Order statistics can be easily combined with descriptive statistics in the univariate case [75]. Higher dimensional variants of these notions are also available and used to represent data visually [77]. An interesting generalization of order statistics to a widely-used topological structure is the contour boxplot [99].

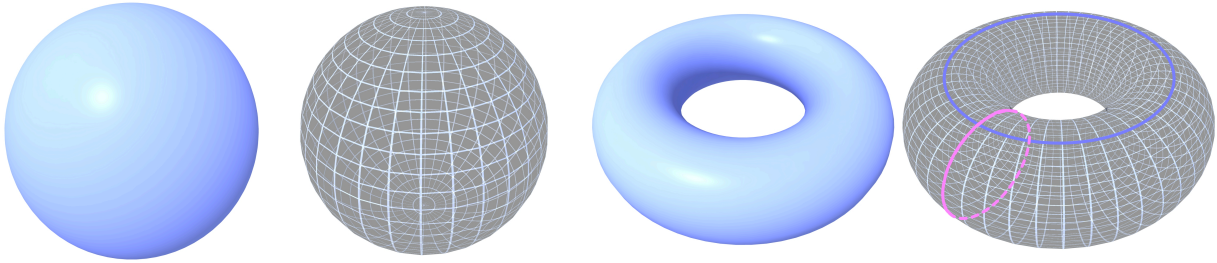


Figure 9: Betti numbers for the sphere and the torus. $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 1$ for the sphere (left) and $\beta_0 = 1$, $\beta_1 = 2$, and $\beta_2 = 1$ for the torus (right). Image courtesy of Mustafa Hajij.

7 Topological Data Analysis

If you can put it on a necklace, it has a one-dimensional hole. If you can fill it with toothpaste, it has a two-dimensional hole. For holes of higher dimensions, you are on your own.

Evelyn Lamb, math and science writer [62]

For topology in visualization, two key developments from computational topology play an essential role in connecting mathematical theories to practice: first, separating features from noise using *persistent homology*; second, abstracting topological summaries of data using *topological structures* such as Reeb graphs, Morse-Small complexes, Jacobi sets, and their variants.

Topology, homology and Betti numbers

Topology has been one of the most exciting research fields in modern mathematics [52]. It is concerned with the properties of space that are preserved under continuous deformations, such as stretching, crumpling, and bending, but not tearing or gluing [100].

The beginning of topology was arguably marked by Leonhard Euler, who published a paper in 1736 that solved the now famous Königsberg bridge problem. In the paper, titled “*The Solution of a Problem Relating to the Geometry of Position*”, Euler was dealing with “a different type of geometry where distance was not relevant” [70]. Johann Benedict Listing was credited as the first to use the word “topology” in print based on his 1847 work titled “*Introductory Studies in Topology*”; although many of Listing’s topological ideas were borrowed from Carl Friedrich Gauss [70]. Both Listing and Bernhard Riemann studied the *components* and *connectivity* of surfaces. Listing examined connectivity in 3-dimensional Euclidean space, and Enrico Betti extended the idea to n dimensions. Henri Poincaré then gave a rigorous basis to the idea of connectivity in a series of papers “*Analysis situs*” in 1895. He introduced the concept of *homology* and improved upon the precise definition of Betti numbers of a space [70]. In other words, it was Poincaré who “gave topology wings” [52] via the notion of homology.

The original motivation to define homology was that it can be used to tell two objects (a.k.a. topological spaces) apart by examining their holes. This process associates a topological space \mathbb{X} with a sequence of abelian groups called homology groups $H(\mathbb{X})$, which, roughly speaking, count and collate *holes* in a space [40]. Informally, homology groups generalize a common-sense notion of connectivity. They detect and describe the connected components (0-dimensional holes), tunnels (1-dimensional holes), voids (2-dimensional holes), and holes of higher dimensions in the space. The p -th Betti number β_p is the rank of the p -th homology group of \mathbb{X} , $H_p(\mathbb{X})$, and captures the number of p -dimensional holes of a topological space. For instance, a sphere contains no tunnels but a void, and a torus contains two tunnels (see Fig. 9).

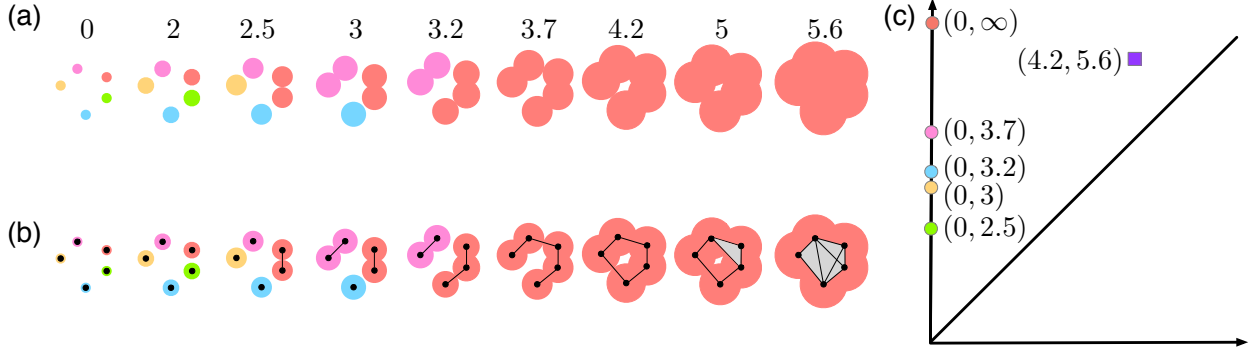


Figure 10: Computing persistent homology of a point cloud in \mathbb{R}^2 . (a) A nested sequence of topological spaces formed by unions of balls at increasing parameter values. (b) A filtration of simplicial complexes that capture the same topological information as in (a). (c) 0- (circles) and 1-dimensional (squares) features in a persistence diagram.

From homology to persistent homology

For simplicity, we work with data represented by simplicial complexes denoted by \mathbb{X} . In algebraic terms, the construction of homology groups begins with a chain complex $C(\mathbb{X})$ that encodes information about \mathbb{X} , which is a sequence of abelian groups $C_0(\mathbb{X}), C_1(\mathbb{X}), \dots$ connected by homomorphisms known as the boundary operators $\partial_p : C_p(\mathbb{X}) \rightarrow C_{p-1}(\mathbb{X})$. The p -th homology group is defined as $H_p(\mathbb{X}) = \ker(\partial_p) / \text{im}(\partial_{p+1})$. The p -th Betti number is the rank of this group, $\beta_p = \text{rank } H_p$, see [67] for an introduction.

Persistent homology transforms the algebraic concept of homology into a multi-scale notion by constructing an extended series of homology groups. In its simplest form, persistent homology applies a homology functor to a sequence of topological spaces connected by inclusions, called a *filtration*. Consider a finite sequence of simplicial complexes connected by inclusions $f_p^{i,j} : \mathbb{X}_i \hookrightarrow \mathbb{X}_j$,

$$\emptyset = \mathbb{X}_0 \hookrightarrow \mathbb{X}_1 \hookrightarrow \dots \hookrightarrow \mathbb{X}_n = \mathbb{X}.$$

Applying p -th homology to this sequence results in a sequence of homology groups connected from left to right by homomorphisms induced by the inclusions,

$$0 = H_p(\mathbb{X}_0) \rightarrow H_p(\mathbb{X}_1) \rightarrow \dots \rightarrow H_p(\mathbb{X}_n) = H_p(\mathbb{X})$$

for each dimension p . The p -th persistent homology group is the image of the homomorphism induced by inclusion, $H_p^{i,j} = \text{im } f_p^{i,j}$ for $0 \leq i \leq j \leq n$. The corresponding p -th persistent Betti number is the rank of this group, $\beta_p^{i,j} = \text{rank } H_p^{i,j}$ [24, Page 151]. As the index increases, the rank of the homology groups changes. When the rank increases (i.e., $f_p^{i-1,i}$ is not surjective), we call this a *birth* event at \mathbb{X}_i ; when the rank decreases (i.e., $f_p^{j-1,j}$ is not injective), we call this a *death* event at \mathbb{X}_j . Persistent homology pairs the birth and the death events as a multi-set of points in the plane called the *persistence diagrams* [29]; see [30, 31] for a comprehensive mathematical introduction. A celebrated theorem of persistent homology is the *stability* of persistence diagrams [16], that is, small changes in the data lead to small changes in the corresponding diagrams, making it suitable for robust data analysis. See Fig. 10 for an example in \mathbb{R}^2 . Given a set of points in \mathbb{R}^2 , we compute its persistent homology by studying the union of balls centered around the points as the radius increases. Here, a green component is born at time 0 and dies when it merges with a red component at time 2.5, resulting a point $(0, 2.5)$ in the persistence diagram. A tunnel is born at time 4.2 and dies at time 5.6, giving rise to a point $(4.2, 5.6)$ in the persistence diagram.

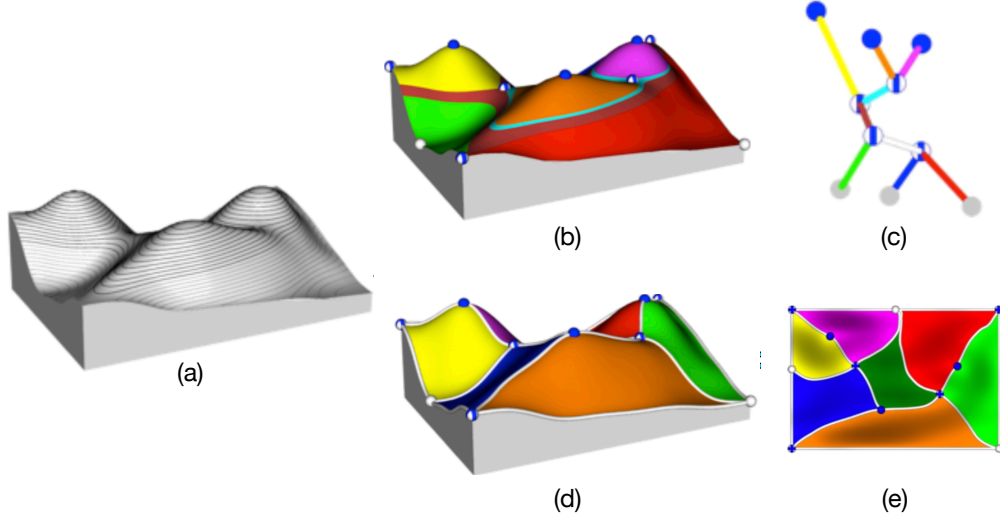


Figure 11: Contour-based (c) and gradient-based (e) topological structures of a 2-dimensional scalar function (a).

Topological structures

Several techniques in topological data analysis and visualization construct topological structures from well-behaved functions on point clouds as summaries of data. On one hand, the well-behavedness is formalized with the Morse theory. On the other hand, such topological structures can be roughly classified into two types: contour-based (Reeb graphs [79], Reeb spaces [27], contour trees [11] and merge trees), and gradient-based topological structures (Morse-Smale complexes [25, 28] and Jacobi sets [23]), see Fig. 11. All such topological structures provide meaningful abstractions of (potentially high-dimensional) data, reduce the amount of data needed to be processed or stored, utilize sophisticated hierarchical representations that capture features at multiple scales, and enable progressive simplifications [63].

MORSE FUNCTION. Let \mathbb{M} be a smooth, compact, and orientable d -manifold without boundary ($d \geq 2$). Suppose \mathbb{M} is equipped with a Riemannian metric so that gradients are well defined. Given a smooth function $f : \mathbb{M} \rightarrow \mathbb{R}$, a point $x \in \mathbb{M}$ is called a *critical point* if the gradient of f at x equals zero, that is, $\nabla f(x) = 0$, and the value of f at x is called a *critical value*. All other points are *regular points* with their function values being *regular values*. A critical point is *non-degenerate* if the Hessian, i.e., the matrix of second partial derivatives at the point, is invertible. A smooth function f is a *Morse function* if (a) all its critical points are non-degenerate; and (b) all its critical values are distinct [24, Page 128]. A pair of two Morse functions is *generic* if their critical points do not overlap.

MORSE-SMALE COMPLEXES. Given a Morse function $f : \mathbb{M} \rightarrow \mathbb{R}$, at any regular point x the gradient is well-defined and integrating it in both directions traces out an integral line, $\gamma : \mathbb{R} \rightarrow \mathbb{M}$, which is a maximal path whose tangent vectors agree with the gradient [28]. Each integral line begins and ends at critical points of f . The *ascending/descending manifolds* of a critical point x is defined as all the points whose integral lines start/end at x . The descending manifolds form a complex called a *Morse complex* of f and the ascending manifolds define the Morse complex of $-f$. The set of intersections of ascending and descending manifolds creates the *Morse-Smale complex* of f . Each *cell* of the Morse-Smale complex is a union of integral lines that all share the same origin and the same destination. In other words, all the points inside a single cell have uniform gradient flow behavior. These cells yield a decomposition into monotonic, non-overlapping regions of the domain, as shown in Fig. 11(b) for a 2-dimensional height function.

JACOBI SET FOR A PAIR OF MORSE FUNCTIONS. Given a generic pair of Morse functions, $f, g : \mathbb{M} \rightarrow \mathbb{R}$, their Jacobi set $\mathbb{J} = \mathbb{J}(f, g) = \mathbb{J}(g, f)$ is the set of points where their gradients are parallel or zero [23].

That is, for some $\lambda \in \mathbb{R}$,

$$\mathbb{J} = \{x \in \mathbb{M} \mid \nabla f(x) + \lambda \nabla g(x) = 0 \text{ or } \nabla g(x) + \lambda \nabla f(x) = 0\}. \quad (19)$$

The sign of λ for each x is called its *alignment*, as it defines whether the two gradients are aligned or anti-aligned. By definition, the Jacobi set contains the critical points of both f and g .

There exist several other descriptions of Jacobi sets [23, 26, 69]. One particularly useful description is in terms of the *comparison measure*, κ [26], which is a gradient-based metric to compare two functions. It plays a significant role in assigning an importance value to subsets of a Jacobi set in terms of the underlying functions f and g by measuring the relative orientation of their gradients.

REEB GRAPHS AND CONTOUR TREES. Let $f : \mathbb{X} \rightarrow \mathbb{R}^d$ be a generic, continuous mapping. Two points $x, y \in \mathbb{X}$ are *equivalent*, denoted by $x \sim y$, if $f(x) = f(y)$ and x and y belong to the same path-connected component of the pre-image of f , $f^{-1}(f(x)) = f^{-1}(f(y))$. The *Reeb space*, $\mathcal{R}(X, f) = \mathbb{X} / \sim$, is the quotient space contained by identifying equivalent points together with the quotient topology inherited from \mathbb{X} . A powerful analysis tool, the *Reeb graph*, is a special case when $d = 1$.

The Reeb graph of a real-valued function $f : \mathbb{X} \rightarrow \mathbb{R}$ describes the connectivity of its level sets. A *contour tree* is a special case of the Reeb graph if the domain \mathbb{X} is simply connected, see Fig. 11(c). A *merge tree* is similar to the Reeb graphs and contour trees except that it describes the connectivity of sublevel sets rather than level sets. The Reeb graph stores information regarding the number of components at any function value as well as how these components split and merge as the function value changes. Such an abstraction offers a global summary of the topology of the level sets and connects naturally with visualization.

8 Color spaces

Although many great thinkers have held that an analytical or mathematical treatment of the subject is impossible or even undesirable, they have gradually deserted the field so that today and indeed throughout the past 50 years it has been generally recognized that a theory of color perception must be, both in form and content, a mathematical theory.

Howard L. Resnikoff, mathematician and business executive [81]

Color is one of the central aspects of visualization and against common belief, a surprisingly mathematical one. Operations on color are an important aspect in many applications, e.g., color mapping, re-sampling of color images or movies, and image manipulations, such as stitching, morphing, or contrast adaption. These operations can be expressed through mathematical formulae if the colors themselves can be expressed as elements of a mathematical space, in which certain concepts such as sums or distances have a meaning. However, as we will see, this is not easy.

The space of all colors is in principal infinite-dimensional because any function over the frequencies of the visible spectrum forms a color. Since, however, the human eye has only three receptors for color, the space of distinguishable colors for humans is only three-dimensional [44, 95]. Depending on the choice of the three basis dimensions, many different colorspaces were developed. In displays, the basic colors are usually red, green, and blue (RGB) and for printing, the standard is cyan, magenta, yellow, and key black (CMYK). The XYZ space by the Commission Internationale de L'Eclairage (CIE) is considered as the basis of all modern color spaces [45, 49]. It embeds all visible colors unambiguously into one space of three imaginary primaries [34, 4]. The chromaticity diagram in Fig. 12 is the result of projecting XYZ to the Maxwell triangle $x + y + z = 1$, which forms a representation of all visible hues and saturations.

A number of spaces, e.g., CIELAB, CIELUV, and DIN99, CIECAM [49, 9], were defined as transformations of XYZ to derive an ideal color space [55], where the Euclidean distance is proportional to the perceived color difference.

Human color perception has been known for a while to be non-Euclidean due to the principle called *hue superimportance* [54] (cf. Fig. 13). It refers to the fact that changes in hue are perceived more strongly than changes in saturation. The circumference of a circle of constant luminance and saturation would be

estimated to measure about 4π for its radius, which cannot be embedded in a Euclidean plane. Please note that the length $l \in \mathbb{R}$ of a path $c : \mathbb{R} \rightarrow C$ is defined for arbitrary metric spaces C

$$l = \sup_{0=t_0, \dots, t_n=1} \sum_{i=1}^k \Delta E(c(t_i), c(t_{i+1})). \quad (20)$$

Therefore, classic descriptions of color spaces, such as those of von Helmholtz [95], Schrödinger [82], and Stiles [104], are based on Riemannian manifolds.

However, state-of-the-art research indicates that human color perception is also non-Riemannian, due to the further principle of *diminishing returns* [54], see Fig. 13. In this context, diminishing returns refers to the phenomenon that when presented with two colors A and C and their perceived middle (average/mixture) B , an observer usually judges the sum of the perceived differences of each half greater than the difference of the two outer colors $\Delta(A,B) + \Delta(B,C) > \Delta(A,C)$. This effect is produced by a natural contrast enhancement filter employed into the human perceptual system to adapt to different viewing conditions. This property is dependent upon the distance between colors, especially for large distances.

As a result, modern color difference formulas (e.g., CIEDE1994, CIEDE2000) that were designed to match experimental data produce complicated spaces, which come with challenges. For example, they are not metric spaces. Being a metric is a very basic mathematical property that we would expect from a distance measure $d : C \times C \rightarrow \mathbb{R}$, i.e., that it suffices non-negativity $d(x,y) \geq 0$, identity of indiscernible $d(x,y) = 0 \Leftrightarrow x = y$, symmetry $d(x,y) = d(y,x)$, and the triangle inequality $d(x,z) \leq d(x,y) + d(y,z)$. The reasons for such a challenge are not in the experimental data but can be found in the mathematical models underlying the distance formulae [65, 64, 48]. An example of the violation of the triangle inequality is shown in Fig. 14.

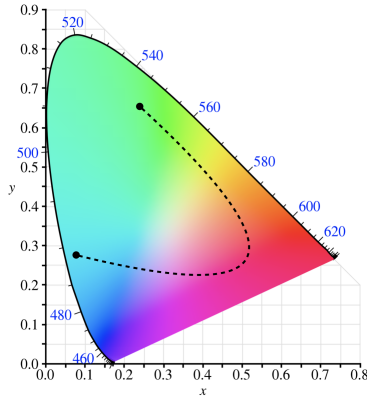


Figure 12: CIE XYZ chromaticity diagram and a path that represents a colormap.

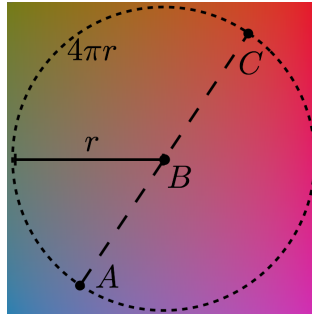


Figure 13: Illustration of hue-superimportance with circumference of $\approx 4\pi r$ and diminishing returns ($\overline{AB} + \overline{BC} > \overline{AC}$).

$$\begin{aligned} \Delta E_{2000}(\text{green, green}) &= 4 \\ \Delta E_{2000}(\text{blue, green}) &= 85 \\ \Delta E_{2000}(\text{blue, yellowish green}) &= 113 \end{aligned}$$

Figure 14: Illustration of non-metric behavior of CIE ΔE_{2000} . Violation of the triangle inequality implies that the path over green RGB=(146,252,77) is shorter than the direct path from blue RGB=(0,0,255) to yellowish green RGB=(177,253,79), which is very counter intuitive.

The difficulties, however, lie not only in the modeling of the color spaces but also in the visualization side. Mathematical operations on color become significantly harder in non-Euclidean spaces. As a basic example, consider linear interpolation where values are taken equidistantly on a straight line connecting two points. In non-Euclidean spaces, the concept of a straight line is, in general, undefined.

To overcome some of these difficulties, some authors generate spaces that are close to the original distance measure but are Euclidean or at least Riemannian [94, 78]. This, however, conflicts with the experimental results from the perceptual sciences. We believe that future color spaces will continue to better approximate human color perception and embrace its complicated non-Euclidean structure because our computational capacities will enable us to work with them despite those difficulties. We believe that the path forward lies in improving visualization algorithms so that they run on general non-Euclidean color spaces. A few results been obtained recently for color interpolation [105] and colormap assessment [8].

Acknowledgement

The authors would like to thank the organizers of Dagstuhl Seminar 18041 titled “Foundations of Data Visualization” in 2018. BW is partially supported by NSF IIS-1910733, DBI-1661375, and IIS-1513616. RB is partially supported by the Laboratory Directed Research and Development (LDRD) program of Los Alamos National Laboratory (LANL) under project number 20190143ER. IH is supported through Swedish e-Science Research Center (SeRC) and the ELLIIT environment for strategic research in Sweden.

References

- [1] H. Amann. *Ordinary Differential Equations: An Introduction to Nonlinear Analysis*, volume 13 of *Studies in Mathematics*. Walter de Gruyter, 2011.
- [2] M. Berger. *A Panoramic View of Riemannian Geometry*. Springer, 2003.
- [3] J. Böttger, A. Schäfer, G. Lohmann, A. Villringer, and D. S. Margulies. Three-dimensional mean-shift edge bundling for the visualization of functional connectivity in the brain. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):471–480, March 2014.
- [4] A. Broadbent. Calculation from the original experimental data of the CIE 1931 RGB standard observer spectral chromaticity coordinates and color matching functions. *Québec, Canada: Département de génie chimique, Université de Sherbrooke*, 2008.
- [5] A. Brun, H. Knutsson, H.-J. Park, M. E. Shenton, and C.-F. Westin. Clustering fiber traces using normalized cuts. In C. Barillot, D. R. Haynor, and P. Hellier, editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 368–375. Springer Berlin Heidelberg, 2004.
- [6] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.
- [7] R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzler. Moment invariants for 2D flow fields via normalization in detail. *IEEE Transactions on Visualization and Computer Graphics*, 21(8):916–929, Aug 2015.
- [8] R. Bujack, T. L. Turton, F. Samsel, C. Ware, D. H. Rogers, and J. Ahrens. The good, the bad, and the ugly: A theoretical framework for the assessment of continuous colormaps. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):923–933, Jan 2018.
- [9] H. Büring. Eigenschaften des farbenraumes nach din 6176 (din99-formel) und seine bedeutung für die industrielle anwendung. In *Proceedings of 8th Workshop Farbbildverarbeitung der German Color Group*, pages 11–17, 2002.
- [10] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.
- [11] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.
- [12] M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200 – 210, 2013.
- [13] C. Chen, C. Wang, K. Ma, and A. T. Wittenberg. Static correlation visualization for large time-varying volume data. In *IEEE Pacific Visualization Symposium*, pages 27–34, 2011.
- [14] E. A. Coddington. *An introduction to ordinary differential equations*. Courier Corporation, 2012.
- [15] J. G. Coffin. *Vector analysis: An introduction to vector-methods and their various applications to physics and mathematics*. J. Wiley & Sons, 1911.
- [16] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007.
- [17] C. D. Correa and P. Lindstrom. Towards robust topology of sparsely sampled data. *Transactions on Computer Graphics and Visualization*, 17(12):1852–1861, 2011.

- [18] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer, 3rd edition, 2008.
- [19] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In *ACM SIGGRAPH 2006 Courses*, pages 39–54. ACM, 2006.
- [20] M. Desbrun, K. Polthier, P. Schröder, and A. Stern. Discrete differential geometry. In *ACM SIGGRAPH 2006 Courses*, page 1. ACM, 2006.
- [21] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [22] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2001.
- [23] H. Edelsbrunner and J. Harer. Jacobi sets of multiple Morse functions. In F. Cucker, R. DeVore, P. Olver, and E. Süli, editors, *Foundations of Computational Mathematics, Minneapolis 2002*, pages 37–57. Cambridge University Press, 2002.
- [24] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
- [25] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. *Proceedings of the 19th ACM Symposium on Computational Geometry*, pages 361–370, 2003.
- [26] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Local and global comparison of continuous functions. In *IEEE Visualization*, pages 275–280, Oct 2004.
- [27] H. Edelsbrunner, J. Harer, and A. K. Patel. Reeb spaces of piecewise linear mappings. In *Proceedings of the 24th Annual Symposium on Computational Geometry*, pages 242–250. ACM, 2008.
- [28] H. Edelsbrunner, J. Harer, and A. J. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.
- [29] H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.
- [30] H. Edelsbrunner and D. Morozov. Persistent homology: Theory and practice. *European Congress of Mathematics*, 2012.
- [31] H. Edelsbrunner and D. Morozov. Persistent homology. In J. E. Goodman, J. O’Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, Discrete Mathematics and Its Applications, chapter 24. CRC Press LLC, 2017.
- [32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.
- [33] V. Estivill-Castro. Why so many clustering algorithms: A position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, June 2002.
- [34] H. S. Fairman, M. H. Brill, H. Hemmendinger, et al. How the CIE 1931 color-matching functions were derived from wright-guild data. *Color Research & Application*, 22(1):11–23, 1997.
- [35] G. Farin. *Curves and Surfaces for CAD: A Practical Guide*. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann Publishers, 5th edition, 2002.
- [36] G. B. Folland. *Introduction to Partial Differential Equations*. Princeton University Press, 2nd edition, 1995.
- [37] R. Forman. A user’s guide to discrete Morse theory. *Séminaire Lotharingien de Combinatoire*, 48, 2002.
- [38] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.

- [39] H.-O. Georgii. *Stochastics: Introduction to Probability and Statistics*. De Gruyter, 2008.
- [40] R. Ghrist. Three examples of applied and computational homology. *Nieuw Archief voor Wiskunde (The Amsterdam Archive, Special issue on the occasion of the fifth European Congress of Mathematics)*, pages 122–125, 2008.
- [41] A. S. Glassner. *Principles of Digital Image Synthesis*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers Inc., 1995.
- [42] R. Goodman and N. R. Wallach. *Symmetry, Representations, and Invariants*. Number 255 in Graduate Texts in Mathematics. Springer, 2009.
- [43] L. Gosink, J. Anderson, W. Bethel, and K. Joy. Variable interactions in query-driven visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1400–1407, Nov 2007.
- [44] H. Grassmann. Zur Theorie der Farbenmischung. *Annalen der Physik*, 165(5):69–84, 1853.
- [45] J. Guild. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London. Series A*, 230:149–187, 1932.
- [46] C. D. Hansen, M. Chen, C. R. Johnson, A. E. Kaufman, and H. Hagen, editors. *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*. Mathematics and Visualization. Springer, 2014.
- [47] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [48] R. Huertas, M. Melgosa, and C. Oleari. Performance of a color-difference formula based on OSA-UCS space using small-medium color differences. *JOSA A*, 23(9):2077–2084, 2006.
- [49] International Commission on Illumination. *Colorimetry*. CIE technical report. Commission Internationale de l’Eclairage, 2004.
- [50] C. Y. Ip, A. Varshney, and J. JaJa. Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2355–2363, Dec 2012.
- [51] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, Sept. 1999.
- [52] I. M. James, editor. *History of Topology*. Elsevier B.V., 1999.
- [53] J. Jankowai and I. Hotz. Feature level-sets: Generalizing iso-surfaces to multi-variate data. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2018.
- [54] D. B. Judd. Ideal color space: Curvature of color space and its implications for industrial color tolerances. *Palette*, 29(21-28):4–25, 1968.
- [55] D. B. Judd. Ideal color space. *Color Engineering*, 8(2):37, 1970.
- [56] D. Jungnickel. *Graphs, Networks and Algorithms*. Algorithms and Computation in Mathematics. Springer, 4th edition, 2012.
- [57] J. Kasten, J. Reininghaus, I. Hotz, H.-C. Hege, B. R. Noack, G. Daviller, and M. Morzynski. Acceleration feature points of unsteady shear flows. *Archives of Mechanics*, 68(1):55–80, 2016.
- [58] G. Kindlmann and C. Scheidegger. An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2014.
- [59] A. Kratz, C. Auer, M. Stommel, and I. Hotz. Visualization and analysis of second-order tensors: Moving beyond the symmetric positive-definite case. *Computer Graphics Forum - State of the Art Reports*, 32(1):49–74, 2013.
- [60] A. Kratz, B. Meyer, and I. Hotz. A Visual Approach to Analysis of Stress Tensor Fields. In H. Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 188–211. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.
- [61] W. Kühnel. *Differential Geometry: Curves - Surfaces - Manifolds*. Student Mathematical Library. American Mathematical Society, 2015.

- [62] E. Lamb. What we talk about when we talk about holes. Scientific American Blog Network, December 2014.
- [63] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017.
- [64] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.
- [65] M. Mahy, L. Eycken, and A. Oosterlinck. Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Research & Application*, 19(2):105–121, 1994.
- [66] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, 2005.
- [67] J. R. Munkres. *Elements of algebraic topology*. CRC Press Taylor & Francis Group, 1984.
- [68] T. Munzner. *Visualization Analysis & Design*. CRC Press Taylor & Francis Group, 2014.
- [69] S. Nagaraj and V. Natarajan. Simplification of Jacobi sets. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors, *Topological Data Analysis and Visualization: Theory, Algorithms and Applications*, Mathematics and Visualization, pages 91–102. Springer, 2011.
- [70] J. J. O’Connor and E. F. Robertson. *A History of Topology*. MacTutor History of Mathematics, 1996.
- [71] T. Peacock, G. Froyland, and G. Haller. Introduction to focus issue: Objective detection of coherent structures. *Chaos*, 25, 2015.
- [72] R. Peikert and M. Roth. The “parallel vectors” operator - a vector field visualization primitive. *Proceedings of IEEE Visualization*, 14(16):263–270, 1999.
- [73] K. Polthier and M. Schmies. Straightest geodesics on polyhedral surfaces. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, page 391. Springer Verlag, 1998.
- [74] F. H. Post, F. J. Post, T. V. Walsum, and D. Silver. Iconic techniques for feature visualization. In *Proceedings of the 6th Conference on Visualization*, page 288. IEEE Computer Society, 1995.
- [75] K. Potter. *The Visualization of Uncertainty*. PhD thesis, University of Utah, 2010.
- [76] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 1992.
- [77] M. Raj. *Depth-based visualizations for ensemble data and graphs*. PhD thesis, University of Utah, 2018.
- [78] D. Raj Pant and I. Farup. Riemannian formulation and comparison of color difference formulas. *Color Research & Application*, 37(6):429–440, 2012.
- [79] G. Reeb. Sur les points singuliers d’une forme de pfaff complètement intergrable ou d’une fonction numérique. *Comptes Rendus Acad. Science Paris*, 222:847–849, 1946.
- [80] M. Renardy and R. C. Rogers. *An introduction to Partial Differential Equations*, volume 13. Springer Science & Business Media, 2006.
- [81] H. L. Resnikoff. Differential geometry and color perception. *Journal of Mathematical Biology*, 1(2):97–131, 1974.
- [82] E. Schrödinger. Grundlinien einer Theorie der Farbenmetrik im Tagessehen. *Annalen der Physik*, 368(22):481–520, 1920.
- [83] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3):19:1–19:21, July 2017.
- [84] J. Schwichtenberg. *Physics from Symmetry*. Undergraduate Lecture Notes in Physics. Springer, 2nd edition, 2017.

- [85] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.
- [86] P. Shirley. *Fundamentals of Computer Graphics*. AK Peters, Ltd., 2005.
- [87] A. D. Snider and H. F. Davis. *Introduction to Vector Analysis*. William C. Brown, 7th edition, 1987.
- [88] M. Sonka, V. Hlavac, and R. Bohle. *Image processing, analysis and machine vision*. Thomson, 3rd edition, 2008.
- [89] J. Steward. *Multivariate Calculus*. Brooks/Cole CENGAGE Learning, 7th edition, 2019.
- [90] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition, 2016.
- [91] N. Suthar, I. jeet Rajput, and V. kumar Gupta. A technical survey on DBSCAN clustering algorithm. *International Journal of Scientific and Engineering Research*, 4(5), 2013.
- [92] A. C. Telea. *Data Visualization: Principles and Practice*. AK Peters, Ltd., 2nd edition, 2015.
- [93] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.
- [94] P. Urban, M. R. Rosen, R. S. Berns, and D. Schleicher. Embedding non-Euclidean color spaces into euclidean color spaces with minimal isometric disagreement. *Journal of the Optical Society of America A*, 24(6):1516–1528, 2007.
- [95] H. Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867.
- [96] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. Visualization and visual analysis of ensemble data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2018.
- [97] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete Laplace operators: No free lunch. In *Proc. Eurographics Symposium on Geometry processing*, pages 33–37, 2007.
- [98] H. Weyl. *Symmetry*. Princeton University Press, 1952.
- [99] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, Dec. 2013.
- [100] Wikipedia contributors. Topology. Wikipedia, The Free Encyclopedia, March 2018.
- [101] P. C. Wong, H. Foote, R. Leung, D. Adams, and J. Thomas. Data signatures and visualization of scientific data sets. *IEEE Computer Graphics and Applications*, 20(2):12–15, March 2000.
- [102] J. Woodring and H. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–137, Jan 2009.
- [103] M. Q. Y. Wu, R. Faris, and K. Ma. Visual exploration of academic career paths. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 779–786, Aug 2013.
- [104] G. Wyszecki and W. S. Stiles. *Color Science*, volume 8. Wiley New York, 1982.
- [105] M. Zeyen, T. Post, H. Hagen, J. Ahrens, D. Rogers, and R. Bujack. Color interpolation for non-Euclidean color spaces. In *IEEE Scientific Visualization Conference Short Papers*. IEEE, 2018.
- [106] Y. J. Zhang. *Geometric Modeling and Mesh Generation from Scanned Images*. CRC Press Taylor & Francis Group, 2016.