54 PATTERN RECOGNITION

Joseph O'Rourke and Godfried T. Toussaint

INTRODUCTION

The two fundamental problems in a pattern recognition system are feature extraction (shape measurement) and classification. The problem of extracting a vector of shape measurements from a digital image can be further decomposed into three subproblems. The first is the image segmentation problem, i.e., the separation of objects of interest from their background. The cluster analysis methods discussed in Section 54.1 are useful here. The second subproblem is that of finding the objects in the segmented image. An example is the location of text lines in a document as illustrated in Section 54.2. The final subproblem is extracting the shape information from the objects detected. Here there are many tools available depending on the properties of the objects that are to be classified. The Hough transform (Section 54.2), polygonal approximation (Section 54.3), shape measurement (Section 54.4), and polygon decomposition (Section 54.6), are some of the favorite tools used here. Important to many of these tasks is finding a nice viewpoint from which extraction is robust and efficient (Section 54.5). Proximity graphs, discussed in Section 54.2, are used extensively for both cluster analysis and shape measurement.

The classification problem involves the design of efficient decision rules with which to classify the feature vector. The most powerful decision rules are the nonparametric rules which make no assumptions about the underlying distributions of the feature vectors. Of these the nearest-neighbor (NN) rule, treated in Section 54.7, is the most well known. This section covers the three main issues concerning NN-rules: how to edit the data set so that little storage space is used, how to search for the nearest neighbor of a vector efficiently, and how to estimate the future performance of a rule both reliably and efficiently.

54.1 CLUSTER ANALYSIS AND CLASSIFICATION

GLOSSARY

Cluster analysis problem: Partitioning a collection of n points in some fixed-dimensional space into $m < n$ groups that are "natural" in some sense. Here m is usually much smaller than n.

Image segmentation problem: Partitioning the pixels in an image into "meaningful" regions, usually such that each region is associated with one physical object.

Block-based segmentation: Image segmentation based on characteristics of rectangular blocks of pixels. Often contrasted with layer-based segmentation.

Dendrogram: A tree representing a hierarchy of categories or clusters.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Hierarchical clustering algorithms: Those that produce a dendrogram whose root is the whole set and whose leaves are the individual points.

Graph-theoretic clustering: Clustering based on deleting edges from a proximity graph.

K-means clustering: Tracking clusters over time by comparing new data with old means.

Data mining: Algorithmic processing of large (and often unstructured) data sets to uncover previously unknown, informative patterns.

Classical cluster analysis requires partitioning points into natural clumps. "Natural" may mean that the clustering agrees with human perception, or it may simply optimize some natural mathematical measure of similarity or distance so that points that belong to one cluster are similar to each other and points far away from each other are assigned to different clusters. It is not surprising that such a general and fundamental tool has been applied to widely different subproblems in pattern recognition. One obvious application is to the determination of the number and description of classes in a pattern recognition problem where the classes are not known a priori, such as disease classification in particular or taxonomy in general. In this case m is not known beforehand and the cluster analysis reveals it.

A fundamental problem in pattern recognition of images is the segmentation problem: distinguishing the figure from the background. Clustering is one of the most powerful approaches to image segmentation, applicable even to complicated images such as those of outdoor scenes. In this approach each pixel in the $N \times N$ image is treated as a complicated object by associating it with a local neighborhood. For example, we may define the 5×5 neighborhood of pixel p_{ij}, denoted by $N_5[p_{ij}]$, as $\{p_{mn} \mid i-2 \leq m \leq i+2, j-2 \leq n \leq j+2\}$. We next measure k properties of p_{ij} by making k measurements in $N_5[p_{ij}]$. Such measurements may include various moments of the intensity values (grey levels) found in $N_5[p_{ij}]$, etc. Thus each pixel is mapped into a point in k-dimensional pixel-space. Performing a cluster analysis of all the resulting $N \times N$ points in pixel-space yields the desired partitioning of the pixels into categories. Labeling each category of pixels with a different color then produces the segmentation. See [BR14] for a survey of traditional and graph-theoretical techniques for image segmentation, as well as combinations of both approaches.

See [Gor96] for a survey of clustering methods that include constraints beyond similarity, or constraints on the topology of the resulting dendrograms; and see [ZA15] for a survey of image segmentation techniques, with an emphasis on block-based methods. See also Section 47.7 of this Handbook.

HIERARCHICAL CLUSTERING

In taxonomy there is no special number m that we want to discover; rather the goal is the production of a *dendrogram* (tree) that grows all the way from one cluster to n clusters and shows us at once how good a partitioning is obtained for any number of clusters between one and n. Such methods are referred to as *hierarchical* methods. They fall into two groups: ***agglomerative*** (bottom-up, merging) and ***divisive*** (top-down, splitting). Furthermore, each of these methods

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

can be used with a variety of distance measures between subsets to determine when to merge or split. Two popular agglomerative clustering algorithms are the ***single-link*** and the ***complete-linkage*** (or farthest-neighbor) algorithm. In the former, cluster similarity is measured by the minimum of the distances between all pairs of elements, one in each subset, whereas in the latter similarity is measured by the maximum pairwise distance. The complete-linkage criterion tends to produce more compact clusters, while the single-link criterion can suffer from chaining [JMF99]. Krznaric and Levcopoulos [KL02] show that a complete-linkage hierarchy can be computed in optimal $O(n \log n)$ time and $O(n)$ space.

GRAPH-THEORETIC CLUSTERING

The most powerful methods of clustering in difficult problems, which give results having the best agreement with human performance, are the graph-theoretic methods [JT92]. The idea is simple: Compute some proximity graph (such as the minimum spanning tree) of the original points. Then delete (in parallel) any edge in the graph that is much longer (according to some criterion) than its neighbors. The resulting forest is the clustering (each tree in this forest is a cluster). Proximity graphs have also been used effectively to design cluster validity tests [PB97], [YL04], and for various specific problems that occur in pattern recognition such as measuring association, outlier detection, and dimensionality reduction [Mar04].

A useful generalization is the *graph-cut* method, which cuts edges to optimize some energy function. The max-cut/min-flow theorem permits energy minimization via maximizing the flow. See [PZZ13] for a survey.

K-MEANS TYPE CLUSTERING

There are many applications where we know that there are exactly k clusters, for example in character recognition. However, because of external factors such as the variations in people's hand-printing over time, or a change in the parameters of a machine due to wear or weather conditions, the clusters must be "tracked" over time. One of the most popular methods for doing this is the ***k-means algorithm***. The k-means algorithm searches for k cluster centroids in $\mathbb{R}^d$ with the property that the mean squared (Euclidean) distance between each of the n points and its nearest centroid ("mean") is minimized [Mac67]. A determining characteristic of this approach is that the number of clusters k is fixed. A typical heuristic starts with an initial partition, computes centers, assigns data points to their nearest center, recomputes the centroids, and iterates until convergence is achieved according to some criterion. Unfortunately, this attractively simple algorithm's performance depends upon the initial partitioning, and in fact can be forced into a suboptimal solution of arbitrarily high approximation ratio. Moreover, it can take exponential (in n) time to converge, even in the plane [Vat11]. Finding the optimal solution is NP-hard in the plane [MNV12]. This led to developing algorithms with performance guarantees. Matoušek achieved an $O(n \log^k n)$ ϵ-approximation algorithm under the assumption that k and d are fixed [Mat00]. This was improved to an $(9 + \epsilon)$-approximation algorithm via a center-swap heuristic with this provable upper

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

bound [KMN+04]. On the other hand, there is some evidence that the exact k-means algorithm, even though exponential, can be implemented to work well in practice for small d [PM99].

A variation on the k-means algorithm permits splitting and merging of clusters. This technique is employed by the ISODATA algorithm (Interactive Self-Organizing Data Analysis Technique) [Jen96].

DISTANCES BETWEEN SETS

A fundamental computational primitive of almost all clustering algorithms is the frequent computation of some distance measure between two sets (subsets) of points. This is especially so in the popular hierarchical methods discussed above. There exists a large variety of distance and more general similarity measures for this purpose. Here we mention a few. Most efficient algorithms for distance between sets apply only in $\mathbb{R}^2$ but some methods extend to higher dimensions; see [Smi00]. Let P and Q be two convex polygons of n sides each. Two distance measures should be distinguished: the minimum *element* distance, the smallest distance between a vertex or edge of P and a vertex or edge of Q, and the minimum *vertex* distance, the minimum distance between a vertex of P and a vertex of Q. The minimum element distance can be computed in $O(\log n)$ time [Ede85]. On the other hand, computation of the minimum vertex distance between P and Q has a linear lower bound. For the case of two nonintersecting convex polygons several different $O(n)$ time algorithms are available, and the same bound can be achieved for crossing convex polygons [Tou84]. For the case when the two polygons do not intersect, there exist additional distance measures that apply to specific applications areas. One such measure is the Grenander distance [Gre76], [Tou14c] defined as the sum of the lengths of the two chords determined by the critical separating supporting lines, less the sum of the lengths of all the edges of the polygons that belong to the inner chains determined by the supporting lines.

Let P and Q be two x-monotonic rectilinear (isothetic) polygonal functions in two dimensions. This setting occurs in several applications such as measuring music similarity in which case the functions act as models of melodies, and the similarity between two melodies P and Q is measured by the area that lies in between the two curves [ÓMa98]. If this distance is minimized under translations of P (with Q fixed) in the x or y directions then the problem falls under the topic of polygonal matching problems [AFL06], [ACH91].

Let R be a set of n red points and B a set of n blue points in the plane. Both the minimum distance and the maximum distance between R and B can be computed in $O(n \log n)$ time. For the latter problem, two algorithms are available. The first [TM82] is simple but does not appear to generalize to higher dimensions. The second [BT83] works by reducing the maximum distance problem between R and B to computing the diameter of 81 convex polygons. These are obtained by computing the convex hulls of the unions of 81 carefully selected subsets of R and B, and then reporting the maximum of these 81 diameters as the maximum distance. These ideas can be extended to obtain efficient algorithms for all dimensions [Rob93]. Therefore, any improvement in high-dimensional diameter algorithms automatically improves maximum-distance algorithms.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Another distance that has applications to machine learning in general, and support vector machines (SVMs) in particular, is the width of the widest empty separating strip between the two sets of points if they are linearly separable (*strong separators*), and otherwise the thinnest strip that minimizes a given number of classification errors (*weak separators*) [Hou93].

DATA MINING

The explosion of the Web has given new impetus to automated methods to uncover informative patterns from huge, often unstructured, data repositories ("big data"). This activity has become known as ***data mining.*** Clustering to discover structures in the data set is a key component of data mining; see [Ber06] for a survey of clustering in data mining. The k-means algorithm and its variants remain popular, not only in geometric domains (e.g., in geological databases [JMF99], celestial databases [PM99], image databases, and so on), but even for text-document clustering [SKK00]. There is some movement in the literature away from point distances for categorical attributes, for which the iterative centroid-based clustering algorithms are often inappropriate. For example, "links" [GRS00] or context-based measures [DM00] are frequently employed, which places this work close to graph-theoretic clustering. A related direction is finding "unusual" strings of `ACTG` characters within the human genome [ABL02] [BC15].

54.2 EXTRACTING SHAPE FROM DOT PATTERNS

HOUGH TRANSFORMS

The ***Hough transform*** was originally proposed (and patented) as an algorithm to detect straight lines in digital images [Lea92]. The method may be used to detect any parametrizable pattern, and has been generalized to locate arbitrary shapes in images. The basic idea is to let each above-threshold pixel in the image vote for the points in the parameter space that could generate it. The votes are accumulated in a quantized version of the parameter space, with high vote tallies indicating detection.

EXAMPLES

1. *Lines.* Let the lines in the (x, y) image plane be parametrized as $y = mx + b$. Then a pixel at (x_0, y_0) is a witness to a line passing through it, that is, an (m, b) pair satisfying $y_0 = mx_0 + b$. Thus, (x_0, y_0) votes for all those (m, b) pairs: the line in parameter space dual to the pixel.
2. *Circles.* Parametrize the circles by their center and radius, (x_c, y_c, r). Then a pixel (x_0, y_0) gives evidence for all the parameter triples on the circular cone in the 3-parameter space with apex at $(x_0, y_0, 0)$ and axis parallel to the r-axis.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

3. *Object location.* Suppose a known but arbitrary shape S is expected to be found in an image, and its most likely location is sought. For translation-only, the parameter space represents the location of some fixed point of S. Each pixel in the image of the right shading or color votes for all those translations that cover it appropriately.

The above approaches are not necessarily optimal for the tasks listed. For example, it was shown in [CT77] that nonuniform (maximum entropy) quantization with ρ-θ (normal-form) parametrization for lines is superior to uniform quantization with m-b (slope-intercept) parametrization. Since earning a patent in 1962 the Hough transform has been generalized in a variety of ways in order to overcome several of its limitations [MC15].

The demands of high-dimensional vote accumulators have engendered the study of *dynamic quantization* of the parameter space, and ***geometric hashing***. This latter technique has features in the image vote for each member of a library of shapes by hashing into a table using the feature coordinates as a key. Each table entry may correspond to several shapes at several displacements, but all receive votes. Geometric hashing has been applied with some success to the *molecular docking* problem [LW88]; see also [MSW02].

Variants of the Hough transform inspired by results in computational geometry have also appeared. In [AK96] two such algorithms are presented and studied with respect to the tradeoff that exists between computational complexity and effectiveness of line detection. They obtain efficient implementations by using the plane-sweep paradigm.

A significant generalization of the technique is proposed in [FO12], which seeks to detect subspace alignments in noisy multidimensional data sets.

More recent applications of the Hough transform and its variants include the detection of lane lines in traffic and transportation science [BK13], [SSSS10], as well as detecting land mines in mine fields [LSC97]. Present challenges in this area include extending the existing methods that have concentrated on binary images, to also work well on color images, and to reduce the computational complexity of the algorithms by developing parallel implementations.

TEXT LINE ORIENTATION INFERENCE

In an automated document analysis system, given a block (paragraph) of text, the text line orientation inference problem consists of determining the location and direction of the lines of text on the page. Almost always these lines are either horizontal (e.g., English) or vertical (e.g., Chinese) The fundamental geometric property that allows this problem to be solved is the fact that according to a universal typesetting convention guided by ease of reading, characters are printed closer together within text lines than between text lines.

One of the most successful, robust, skew-tolerant, simple, and elegant techniques for text line orientation inference was proposed by Ittner [Itt93], and later improved by Bose et al. [BCG98]. Assume that the given text block B consists of n black connected components (characters). The three key steps in Ittner's procedure

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

are: (1) idealize each character by a point, thus obtaining a set S of n points in the plane; (2) construct the Euclidean minimum spanning tree MST(S) of the n points obtained in (1); and (3) determine the text line orientation by analysis of the distribution of the orientations of the edges in MST(S). Step (1) is done by computing the center of the bounding box of each character. Cheriton and Tarjan proposed a simple algorithm for computing the MST of a graph in $O(E)$ time where E is the number of edges in the graph [CT76]. Fortunately there are many graphs defined on S (usually belonging to the class of proximity graphs [JT92]) that have the property that they contain the MST(S) and also have $O(n)$ edges. For these graphs the Cheriton-Tarjan algorithm runs in $O(n)$ time.

The traditional approaches to text detection described above were focused on two-dimensional inputs such as newspapers, magazines, and books, in which the input consisted of binary (black-white) images. More recently attention has shifted to more difficult problems of detecting text in color images of complex three-dimensional scenes. Ye and Doermann [YD15] provide a survey of new approaches to these more challenging problems.

TEXT BLOCK ISOLATION

The text-block isolation problem consists of extracting blocks of text (paragraphs) from a digitized document. By finding the enclosing rectangles around each connected component (character) and around the entire set of characters we obtain a well structured geometric object, namely, a rectangle with n rectangular "holes." This problem is ideally suited to a computational geometric treatment. Here we mention an elegant method that analyzes the empty (white) spaces in the document [BJF90]. This approach enumerates all maximal white rectangles implied by the black rectangles. A white rectangle is called maximal if it cannot be enlarged while remaining outside the black rectangles. Their enumeration algorithm takes quadratic time in the worst case, but a clever heuristic exploiting properties of layouts leads to $O(n)$ expected time.

RELATIVE NEIGHBORHOOD GRAPHS

Relative neighborhood graphs (RNG's), introduced in [Tou80], capture proximity between points by connecting nearby points with a graph edge. The many possible notions of "nearby" (in several metrics) lead to a variety of related graphs, including the *sphere of influence graph* (SIG). We defer to Chapter 32 for definitions and details, and only mention here several applications. The RNG was proposed as an approach to Marr's "primal sketch" in computer vision, and it continues to have application in low-level vision, extracting perceptually meaningful features from images. The RNG, the SIG, and related proximity graphs find use in nearest-neighbor searching, cluster analysis, data mining, and outlier detection. See [Tou05], [Tou14a], and [Tou14b] for further details.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

54.3 POLYGONAL APPROXIMATION

Let $P = (p_1, p_2, \ldots, p_n)$ be a polygonal curve or chain in the plane, consisting of n points p_i joined by line segments $p_i p_{i+1}$. In general P may be closed and self-intersecting. Polygonal curves occur frequently in pattern recognition either as representations of the boundaries of figures or as functions of time representing, e.g., speech. In order to reduce the complexity of costly processing operations, it is often desirable to approximate a curve P with one that is composed of far fewer segments, yet is a close enough replica of P for the intended application. Some methods of reduction attempt smoothing as well. An important instance of the problem is to determine a new curve $Q = (q_1, q_2, \ldots, q_m)$ such that (1) m is significantly smaller than n; (2) the q_j are selected from among the p_i; and (3) any segment $q_j q_{j+1}$ that replaces the chain $q_j = p_r, \ldots, p_s = q_{j+1}$ is such that the distance between $q_j q_{j+1}$ and each p_k, $r \leq k \leq s$, is less than some predetermined error tolerance ω. Different notions of distance, or error criteria, lead to different algorithmic issues. Moreover, for each distance definition, there are two constrained optimization problems that are of interest, Min-# and Min-ϵ.

GLOSSARY

Distance from point p to segment s: Minimum distance from p to any point of s.

Parallel-strip criterion: All the vertices $p_i, \ldots, p_t$ lie in a parallel strip of width 2ϵ whose center line is collinear with $q_j q_{j+1}$ [ET94].

Segment criterion, or Hausdorff measure: For each p_k, $r \leq k \leq s$, the distance from p_k to $q_j q_{j+1}$ is less than ϵ [MO88, CC96].

Min-# problem: Given the error tolerance ϵ, find a curve $Q = (q_1, \ldots, q_m)$ satisfying the constraint such that m is minimum.

Min-ϵ problem: Given m, find a curve $Q = (q_1, \ldots, q_m)$ satisfying the constraint such that the error tolerance is minimized.

Fréchet distance curves: The smallest "leash length" for a master to walk along one curve while a leashed dog walks along the other.

The main research focus has been the Min-# problem under the Hausdorff error measure, also called ϵ-simplification. Near quadratic-time, $O(n^2 \log n)$, algorithms have been achieved for arbitrary polygonal curves [MO88]. The quadratic-time barrier seems difficult to break, so approximation algorithms have been explored, which achieve an error no more than ϵ but compromise on m, the min-#. A popular algorithm is the Douglas-Peucker algorithm, ***iterative endpoint fitting***, which, although easily implemented, does not guarantee performance and could be quadratic in the worst case. Now near-linear performance with a guarantee has been achieved for simplifying monotone curves [AHM+05].

At least three other simplification directions have been pursued:

1. Rather than using the Hausdorff distance, a large amount of recent work has explored guaranteeing a maximum Fréchet distance between the two polygonal curves. Again near-linear approximations are available [AHM+05]. For

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

polygonal chains in dimensions $d \geq 3$, and an $O(n \log n)$ exact algorithm has been achieved [BJW+08].

2. Tracking objects over long periods has led to exploring streaming algorithms, with limited storage. Here $O(1)$ competitive ratios have been achieved for several cases under both the Hausdorff and Fréchet distance measures [ADB+10].

3. The task of polygonal approximation has been given new significance in three dimensions for its importance in simplifying polyhedral models in computer graphics. There is now a module within CGAL for "Triangulated Surface Mesh Simplification." This topic is covered in detail in Chapter 56.

54.4 SHAPE MEASUREMENT AND REPRESENTATION

MEDIAL AXIS

GLOSSARY

Medial axis: The set of points of a region P with more than one closest point among the boundary points ∂P of the region. Equivalently, it is the set of centers of maximal balls, i.e., of balls in P that are themselves not enclosed in another ball in P.

Voronoi diagram: The partition of a polygonal region P into cells each consisting of the points closer to a particular open edge or vertex than to any other.

The *medial* or *symmetric axis* was introduced by Blum [Blu67] to capture biological shape, and it has since found many other applications, for example, to geometric modeling (*offset* computations; see Section 50.2) and to mesh generation [SNT+92] (Section 29.4). It provides a central "skeleton" for an object that has found many uses. It connects to several other mathematical concepts, including the *cut locus* and most importantly, the Voronoi diagram (Chapter 27).

The medial axis of a convex polygon is a tree with each vertex a leaf. For a nonconvex polygon, the medial axis may have a parabolic arc associated with each reflex vertex (Figure 50.1.5). The basic properties of the medial axis were detailed by Lee [Lee82], who showed that the medial axis of a polygon P is just the Voronoi diagram minus the Voronoi edges incident to reflex vertices, and provided an $O(n \log n)$ algorithm for constructing it. After a long search by the community, an $O(n)$ algorithm was obtained [CSW99]. The simplest implementations are, however, quadratic [YR91].

The medial axis has also found much use in image processing, where its digital computation is via *thinning algorithms*. Pioneered by Rosenfeld, these algorithms are very simple and easily parallelized [Cyc94].

The definition of medial axis extends to $\mathbb{R}^d$, and has found application in $\mathbb{R}^3$. Although an exact algorithm is available for the medial axis of (perhaps nonconvex) polyhedra in $\mathbb{R}^3$ [Cul00], there is no implementation that constructs the exact medial axis of an arbitrary semi-algebraic set [ABE09]. For applications, exact

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

medial axes are rarely needed. An approximate medial axis of a set of points in $\mathbb{R}^3$ can be computed directly from the Voronoi diagram, with guaranteed convergence: [DZ04]. See [TDS+16] for an overview of skeletons in $\mathbb{R}^3$, and [AW13] for closely related work on the *straight skeleton* in $\mathbb{R}^3$.

POINT PATTERN MATCHING

Exact point pattern matching is an interesting algorithmic question related to string matching, but pattern recognition applications usually require some type of approximate matching. Two types may be distinguished [AG00]: one-to-one matching, and Hausdorff matching.

GLOSSARY

One-to-one approximate matching: Let two finite sets of points A and B have the same cardinality. One-to-one matching requires finding a transformation (of a given type) of B such that each point of B is mapped to within a distance of ϵ of a matched point of A. The matches are either determined by ***labels*** on the points, or the points are ***unlabeled*** and the match is to be discovered.

Decision problem: Given ϵ, is there such a matching?

Optimization problem: Find the minimum ϵ for which an approximate matching exists.

Hausdorff distance: For two finite sets A and B, perhaps of different cardinalities, the largest of the between-sets nearest-neighbor distances.

Hausdorff matching: Find a transformation of B that minimizes the Hausdorff distance from A.

The most combinatorially interesting point matching (unrealistically) demands exact matching. One version of this is the ***congruent subset detection problem***: Given a pattern set A of m points, find all subsets of a background set B of n points that are congruent to A. Solving this in the plane relies on the unsolved Erdős problem of bounding the number of unit-distance pairs among n points, whose best upper bound is $O(n^{4/3})$ (Chapter 10). Important variations are obtained by acting on the pattern by some group, e.g., translations. Results here are surveyed in [Bra02], from which the results shown in Table 54.4.1 are gathered ($\alpha()$ is the near-constant inverse Ackermann function; cf. Chapter 28).

A window-restricted version of the problem led Brass to pose the following interesting conjecture:

- Any set of n points in the plane contains only $O(n)$ empty congruent triangles.

There are sets with $\binom{n}{3}$ empty triangles.

Results on ***one-to-one approximate matching*** algorithms obtained for a variety of permissible transformations in [AMW+88] are shown in Table 54.4.2.

Hausdorff matching leads to analysis of envelopes of *Voronoi surfaces*. Typical results are shown in Table 54.4.3. Here we show the complexities when $|A| = |B| = n$, although the algorithms work for sets of different cardinalities.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

TABLE 54.4.1 Subset detection of m points among n points.

GROUP	DIM	MATCHES	ALGORITHM
Congruence	2	$O(n^{4/3})$	$O(mn^{4/3}\log n)$
Congruence	3	$\Omega(n^{4/3})$	$O(mn^{5/3}\log n 2^{O(\alpha(n)^2)})$
Translation	d	$n-\Theta(n^{1-1/k})$, $k\le d$	$O(mn\log n)$
Homothets	d	$O(n^{1+1/k})$, $k\le d$	$O(mn^{1+1/d}\log n)$
Similarity	d	$O(n^d)$	$O(mn^d\log n)$
Affine	d	$O(n^{d+1})$	$O(mn^{d+1}\log n)$

TABLE 54.4.2 One-to-one point matching in two dimensions.

MOVEMENTS	LABELED	ϵ	COMPLEXITY
Translation	labeled	dec, opt	$O(n)$
Translation	unlabeled	decision	$O(n^6)$
Translation	unlabeled	optimization	$O(n^6\log n)$
Rotation	labeled	decision	$O(n\log n)$
Rotation	labeled	optimization	$O(n^2)$
Trans+rot+refl	labeled	decision	$O(n^3\log n)$
Trans+rot+refl	unlabeled	decision	$O(n^8)$

TABLE 54.4.3 Hausdorff matching in the L_2 metric.

MOVEMENTS	DIM	COMPLEXITY
Translation	2	$O(n^3\log n)$
Translation + rotation	2	$O(n^6\log n)$
Translation	3	$O(n^{5+\epsilon})$

Another type of matching is *order type matching* (cf. Section 5.2). In [GP83], an $O(n^3)$ algorithm is given for finding all matchings between two planar point configurations in which their order types agree.

SYMMETRY DETECTION

Symmetry is an important feature in the analysis and synthesis of shape and form and has received considerable attention in the pattern recognition and computer graphics literatures. In [WWV85] an $O(n\log n)$ algorithm is presented for computing the rotational symmetries of polygons and polyhedra of n vertices, but the constant in $\mathbb{R}^3$ is very large. Jiang and Bunke [JB91] give a simple and practical $O(n^2)$ time algorithm for polyhedra. One of the earliest applications of computational geometry to symmetry detection was the algorithm of Akl and Toussaint [AT79] to check for polygon similarity. For a survey of the early work on detecting symmetry, see [Ead88]. Since then attention has been given to other aspects of symmetry and for objects other than polygons. Sugihara [Sug84] shows how a modification of the

planar graph-isomorphism algorithm of Hopcroft and Tarjan can be used to find all symmetries of a wide class of polyhedra in $O(n \log n)$ time.

Most previous methods for measuring the symmetries of polygons and polyhedra have focused on a binary output: an object either has or does not have some symmetries. However, for many problems in pattern recognition a more useful measure is a graded (or continuous) descriptor that yields an indication of the amount of symmetry contained in an object. Kazhdan et al. [KCD+04] present an algorithm that computes a reflective symmetry descriptor in $O(n^4 \log n)$ time for an $n \times n \times n$ grid of voxels in $3D$. The approach in [KCD+04] is to measure the amount of reflective symmetry along an axis, through the center of gravity of the pattern, by the amount of overlap between the original pattern and the pattern reflected about that axis. This provides a *global* graded symmetry descriptor. An alternate approach is to measure all *local* reflection symmetries (sub-symmetries) rather than only global ones, and do so with respect to all locations in the object rather than just the center of mass. Such symmetries correlate better with human judgments of complexity than do global symmetries [TOV15].

A related topic is centers of symmetry. Given a convex polygon P, associate with each point p in P the minimum area of the polygon to the left of any chord through p. The maximum over all points in P is known as ***Winternitz's measure of symmetry*** and the point p^* that achieves this maximum is called the ***center of area***. Diaz and O'Rourke [DO94] show that p^* is unique and propose an algorithm for computing p^* in time $O(n^6 \log^2 n)$. See [BT10] for an improved implementation and analysis of the Diaz-O'Rourke algorithm for finding the Simpson point of a convex polygon.

THE ALPHA HULL

The α-shape $\mathcal{S}_\alpha$ of a set S of n points in $\mathbb{R}^3$ is a polyhedral surface whose boundary is a particular collection of triangles, edges, and vertices determined by the points of S [EM94]. It is similar in spirit to the β-skeleton of Section 54.2 in that it is a parametrized collection of shapes determined by an empty balls condition, but it emphasizes the external rather than the internal structure of the set. Let T be a subset of S of 1, 2, or 3 points. Then the convex hull of T, $\text{conv}(T)$, is part of the boundary $\partial\mathcal{S}_\alpha$ of the α-shape iff the surface of some ball of radius α includes exactly the points of T while its interior is empty of points of S. Thus a triangle $\text{conv}(T)$ is part of $\partial\mathcal{S}_\alpha$ iff there is an open α-ball that can "erase" all of the triangle but leave its vertices. $\mathcal{S}_\alpha$ is defined for all $0 \leq \alpha \leq \infty$, with $\mathcal{S}_0 = S$ and $\mathcal{S}_\infty = \text{conv}(S)$.

Every edge and triangle of $\mathcal{S}_\alpha$ is present in the Delaunay triangulation DT of S, and every edge and triangle in DT is present in some $\mathcal{S}_\alpha$. If α is varied continuously over its full range starting from ∞, the convex hull of S is gradually "eaten away" by smaller and smaller α-ball erasers, eventually exposing the original set of points. In between, the α-shape bounds a subcomplex of DT that represents the shape of S.

The alpha shape has been used for cluster analysis, molecular modeling, and the analysis of medical data, among other applications. High-quality code is available: CGAL includes alpha shapes in its basic library (Chapter 68), and a package in R is also available [LPPD16].

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

54.5 NICE VIEWPOINTS AND PROJECTIONS

A robot navigating in 3D space faces a variety of pattern recognition problems that involve classifying objects modeled as polyhedra. A polyhedral object in 3D space is often well represented by a set of points (vertices) and line segments (edges) that act as its features. The feature extraction process involves obtaining *nice* viewpoints of the polyhedron. By a nice viewpoint of an object we mean a projective view in which all (or most) of the features of the object, relevant for some task, are clearly visible. Such a view is often called a nondegenerate view or projection. A recent survey of this topic can be found in [Tou00].

GLOSSARY

Nice viewpoint: A projection of a 3D object (set of points, etc.) onto a plane such that it has some desirable special property.

Knot diagram: A regular projection of a polygon in 3-dimensions onto a plane.

Degeneracies: Properties of objects such as three points collinear.

General position: A configuration of an object such that some specified degeneracies are absent.

Orthogonal projection: A projection from a point at infinity.

Perspective projection: A projection from a point not at infinity.

Robust algorithm: One that works correctly even in the presence of degeneracies.

Regular projection: An orthogonal projection of S such that no three points of S project to the same point on H, and no vertex of S projects to the same point on H as any other point of S.

Wirtinger projections: Regular projections in which no two consecutive edges of the 3D polygon have collinear projections.

Robust nondegenerate projection: A projection that remains nondegenerate even if the object is slightly perturbed.

Decision problem: Given an object and a projection of it, does the projection contain a degeneracy?

Computation problem: Given an object, compute a projection that does not contain a specified degeneracy.

Optimization problem: Given an object, compute the most robust nondegenerate projection.

REGULAR PROJECTIONS

The earliest work on nondegenerate orthogonal projections appears to be in the area of knot theory. Let S be a set of n disjoint line segments in $\mathbb{R}^3$ specified by the Cartesian coordinates of their endpoints (vertices of S) and let H be a plane. Let

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

SH be the orthogonal projection of S onto H. An orthogonal projection of S is said to be *regular* if no three points of S project to the same point on H and no vertex of S projects to the same point on H as any other point of S [Liv93]. This definition implies that for disjoint line segments (1) no point of SH corresponds to more than one vertex of S, (2) no point of SH corresponds to a vertex of S and an interior point of an edge of S, and (3) no point of SH corresponds to more than two interior points of edges of S. Therefore the only crossing points (intersections) allowed in a regular projection are those points that belong to the interiors of precisely two edges of S. This condition is crucial for the successful visualization and manipulation of knots [Liv93].

Regular projections of 3D polygons were first studied by the knot theorist K. Reidemeister [Rei32] in 1932 who showed that all 3D polygons (knots) admit a regular projection, and in fact almost all projections of polygons are regular. Reidemeister however was not concerned with computing regular projections. The computational aspects of regular projections of knots were first investigated by Bose et al., [BGRT99] under the real RAM model of computation. Given a polygonal object (geometric graph, wire-frame or skeleton) in $\mathbb{R}^3$ (such as a simple polygon, knot, skeleton of a Voronoi diagram or solid model mesh), they consider the problem of computing several "nice" orthogonal projections of the object. One such projection, well known in the graph-drawing literature, is a projection with few crossings. They consider the most general polygonal object, i.e., a set of n disjoint line segments, and show that deciding whether it admits a crossing-free projection can be done in $O(n^2 \log n + k)$ time and $O(n^2 + k)$ space, where k is the number of intersections among a set of "forbidden" quadrilaterals on the direction sphere, and $k = O(n^4)$. This implies for example that, given a knot, one can determine if there exists a plane on which its projection is a simple polygon, within the same complexity. Furthermore, if such a projection does not exist, a minimum-crossing projection can be found in $O(n^4)$ time and $O(n^2)$ space. They showed (independently of Reidemeister) that a set of line segments in space (which includes polygonal objects as special cases) always admits a regular projection, and that such a projection can be obtained in $O(n^3)$ time. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k \log n)$ time, where k is the number of intersections of a set of quadratic arcs on the direction sphere and $k = O(n^6)$. Finally, when the objects are polygons and trees in space, they consider monotonic projections, i.e., projections such that every path from the root of the tree to every leaf is monotonic in some common direction on the projection plane. For example, given a polygonal chain P, they can determine in $O(n)$ time if P is monotonic on the projection plane, and in $O(n \log n)$ time they can find all the viewing directions with respect to which P is monotonic. In addition, in $O(n^2)$ time, they can determine all directions for which a given tree or a given simple polygon is monotonic.

COMPUTER VISION

In the computer vision field there is both a theoretical [BWR93] interest in nondegenerate projections and a practical one [DWT99]. The theoretical work resembles the work described in the previous section in that it is assumed that the object

consists of idealized points and line segments or polygons and polyhedra. A tool used for computing viewpoints from which the maximum number of faces of a solid polyhedron is visible, is the ***aspect graph*** [PD90] (Chapter 33).

WIRTINGER PROJECTIONS

That certain types of nondegenerate orthogonal projections of 3D polygons always exist for some directions of projection was rediscovered by Bhattacharya and Rosenfeld [BR94] for a restricted class of regular projections. Those regular projections, in which it is also required that no two consecutive edges of the 3D polygon have collinear projections, are known as ***Wirtinger projections***. Bose et al. [BGRT99] study the complexity of computing a single Wirtinger projection as well as constructing a description of all such projections for the more general input consisting of disjoint line segments. These results include therefore results for 3D chains, polygons, trees and geometric graphs in general. The description of all projections allows one to obtain Wirtinger projections that optimize additional properties. For example, one may be interested in obtaining the most robust projection in the sense that it maximizes the deviation of the viewpoint required to violate the Wirtinger property.

VISUALIZATION

In computer graphics one is interested in visualizing objects well, and therefore *nice* views and nondegenerate views are major concerns. For example, Kamada and Kawai [KK88] proposed a method to obtain nice projections by making sure that in the projection, parallel line segments on a plane in 3D project as far away from each other as possible. Intuitively, the viewer should be as orthogonal as possible to every face of the 3D object. Of course this is not possible and therefore they suggest minimizing (over all faces) the maximum angle deviation between a normal to the face and the line of sight from the viewer. They then propose an algorithm to solve this problem in $O(n^6 \log n)$ time, where n is the number of edges in the polyhedral object in 3D. Gómez et al. [GRT01] reduce this complexity to $O(n^4)$ time. Furthermore, they show that if one is restricted to viewing an object from only a hemisphere, as is the case with a building on top of flat ground, then a further reduction in complexity is possible to $O(n^2)$ time.

A rather different optimization problem regarding projections arises in visualization applications in which it is desired to display visual content on three-dimensional real-world objects rather than on a traditional two-dimensional displays, possibly using more than one projector. The problems here consist of determining optimal placements of the projectors to optimize a variety of objective functions under possible constraints [LAM10].

REMOVING DEGENERACIES

Algorithms in computational geometry are usually designed for the real RAM (random access machine) assuming that the input is in *general position*. More specif-

ically, the general position assumption implies that the input to an algorithm for solving a specific problem is free of certain degeneracies. Yap [Yap90] has distinguished between intrinsic or *problem-induced* and extrinsic or *algorithm-induced* degeneracies (see also Chapter 45). For example, in computing the convex hull of a set of points in the plane, where "left" turns and "right" turns are fundamental primitives, three collinear points constitute a problem-induced degeneracy. On the other hand, for certain vertical line-sweep algorithms, two points with the same x-coordinate constitute an algorithm-induced degeneracy. Computational geometers make these assumptions because doing so makes it not only much easier to design algorithms but often yields algorithms with reduced worst-case complexities. On the other hand, to the implementers of geometric algorithms these assumptions are frustrating. Programmers would like the algorithms to work for any input that they may encounter in practice, regardless of any degeneracies contained in the input.

Often a typical computational geometry paper will make a nondegeneracy assumption that can in fact be removed (*without* perturbing the input) by a global rigid transformation of the input (such as a rotation, for example). Once the solution is obtained on the transformed nondegenerate input, the solution can be transformed back trivially (by an inverse rotation) to yield the solution to the original problem. In these situations, by applying suitable *pre-* and *post*-processing steps, one obtains the *exact* solution to the *original* problem using an algorithm that assumes a nondegenerate input, even when that input is in fact degenerate. This approach not only handles algorithm-induced degeneracies via orthogonal projections but some problem-induced degeneracies as well with the aid of perspective projections.

Gómez et al. [GRT01] consider several nondegeneracy assumptions that are typically made in the literature, propose efficient algorithms for performing the suitable rotations that remove these degeneracies, analyze their complexity in the real RAM model of computation and, for some of these problems, give lower bounds on their worst-case complexity. The assumptions considered in [GRT01] are summarized in Tables 54.5.1 and reftab:degen.segs $\lambda(\cdot)$ is nearly linear; cf. Section 28.10).

PERSPECTIVE PROJECTIONS AND INTRINSIC DEGENERACIES

Intrinsic degeneracies cannot be removed by rotations of the input. If a set of points S in 3D contains three collinear points then so does every orthogonal projection of S. This is where *perspective* projections come to the rescue. However, not all intrinsic degeneracies can be removed with perspective projections. Intrinsic degeneracies that can be removed via perspective projections are called *quasi-intrinsic degeneracies* [HS97, GHS$^+$01].

Gómez et al. [GHS$^+$01] consider computing nondegenerate *perspective* projections of sets of points and line segments in 3D space. For sets of points they give algorithms for computing perspective projections such that (1) all points have distinct x-coordinates, (2) all points have both distinct x- and y-coordinates, (3) no three points in the projection are collinear, and (4) no four points in the projection are cocircular. For sets of line segments they present an algorithm for computing a perspective projection with no two segments parallel. All their algorithms have time and space complexities bounded by low degree polynomials.

TABLE 54.5.1 Removing degeneracies: Point sets.

PROBLEM	DECISION	COMPUTATION	OPTIMIZATION
2D			
No two on vertical line	$\Theta(n \log n)$	$O(n \log n)$	$O(n^2 \log n)$ time, $O(n^2)$ space
			$O(n^2)$ time, space with floor functions
3D			
No two on vertical line	$\Theta(n \log n)$	$O(n \log n)$	$O(n^2 \log n)$ time, $O(n^2)$ space
No two with same x-coordinate	$\Theta(n \log n)$	$O(n \log n)$	$O(n^4)$ time, space
			$O(n^2 \lambda_6(n^2) \log n)$ time, $O(n^2)$ space
No two with same x, y or z-coord	$\Theta(n \log n)$	$O(n \log n)$	OPEN
No three on vertical plane	(3SUM-hard) $O(n^2)$ time, space	(3SUM-hard) $O(n^2)$ time, space	$O(n^6)$ time and space
		$O(n^3)$ time, $O(n)$ space	

TABLE 54.5.2 Removing degeneracies: Line segments and faces.

PROBLEM	DECISION	COMPUTATION	OPTIMIZATION
LINE SEGMENTS			
2D			
No vertical	$\Theta(n)$	$\Theta(n)$	$O(n \log n)$ time, $O(n)$ space
3D			
No vertical	$\Theta(n)$	$\Theta(n)$	$O(n \log n)$ time, $O(n)$ space
No two on vertical plane	$O(n \log n)$	$O(n^2)$ time, $O(n)$ space	$O(n^4)$ time, space
			$O(n^2 \lambda_6(n^2) \log n)$ time, $O(n^2)$ space
FACES			
No face of polyhedron vertical	$\Theta(n)$	$\Theta(n)$	$O(n^2)$ time, space
			$O(n \lambda_6(n) \log n)$ time, $O(n)$ space

FINITE-RESOLUTION MODELS OF VIEW DEGENERACY

View degeneracy is a central concern in robotics where a robot must navigate and recognize objects based on views of the scene at hand [DPR92a, DPR92b]. In the idealized world assumed in the previous sections, degenerate views are not much of a problem if a viewpoint is chosen at random, since almost all projections are not degenerate. On the other hand, real-world digital cameras have a finite resolution and therefore view degeneracy can no longer be ignored [KF87].

OPEN PROBLEMS

A more practical approach would give some thickness to the objects, i.e., consider the points as little balls and the edges of the polyhedra as thin cylinders, and then to redesign the algorithms accordingly. This may turn out to be rather expensive. In practice it may be much more efficient to perform a half-dozen *random* rotations to obtain a nice projection. After all, for many problems in the idealized infinite precision model, a single random rotation yields a nice projection with probability one. Computing optimal projections on the other hand is another matter. Here approximate algorithms may yield efficient solutions that are near-optimal, but these are open problems.

54.6 NEAREST-NEIGHBOR DECISION RULES

GLOSSARY

Nearest-neighbor decision rule: Classifies a feature vector with the closest sample point in parameter space.

In the typical nonparametric classification problem (see Devroye, Gyorfy and Lugosi [DGL96]) we have available a set of d measurements or observations (also called a feature vector) taken from each member of a data set of n objects (patterns) denoted by $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, where X_i and Y_i denote, respectively, the feature vector on the ith object and the class label of that object. One of the most attractive decision procedures is the nearest-neighbor rule (1-*NN*-rule) [FH51]. Let Z be a new pattern (feature vector) to be classified and let X_j be the feature vector in $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ closest to Z. The nearest neighbor decision rule classifies the unknown pattern Z into class Y_j. In the 1960s and 1970s many pattern recognition practitioners resisted using the 1-*NN*-rule on the grounds of the mistaken assumptions that (1) all the data $\{X, Y\}$ must be stored in order to implement such a rule, (2) to determine Y_j, distances must be computed between the unknown vector Z and all the members of $\{X, Y\}$, and (3) such a rule is difficult to implement in parallel using a neural network. Computational geometry research in the 1980s and 1990s along with faster and cheaper hardware has made the *NN*-rules a practical reality [Tou02].

MINIMAL-SIZE TRAINING-SET CONSISTENT SUBSETS

A question that has received a lot of attention in the past fifty years concerns the problem of reducing the number of patterns in the training set $\{X, Y\}$ without degrading the performance of the decision rule. In 1968 Hart was the first to propose an algorithm for reducing the size of the stored data for the nearest neighbor decision rule [Har68]. Hart defined a *consistent* subset of the data as one that classified the entire set correctly with the nearest neighbor rule. He then proposed

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

an $O(n^3)$ time algorithm that he called *CNN* (Condensed Nearest Neighbor) for selecting a consistent subset by heuristically searching for data that were near the decision boundary. However, the method does not in general yield a minimal-size consistent subset.

The first researchers to deal with computing a *minimal-size* consistent subset were Ritter et al. [RWLI75]. They proposed a procedure they called a *selective* nearest neighbor rule (*SNN*) to obtain a minimal-size consistent subset of $\{X, Y\}$, call it S, with one additional property that Hart's *CNN* does not have. Any consistent subset C obtained by *CNN* has the property that every element of $\{X, Y\}$ is nearer to an element in C of the same class than to any element in C of a different class. On the other hand, the consistent subset S of Ritter et al. [RWLI75] has the additional property that every element of $\{X, Y\}$ is nearer to an element in S of the same class than to any element, in the *complete* set, $\{X, Y\}$ of a different class. This additional property of *SNN* tends to keep points closer to the decision boundary than does *CNN*, and allows Ritter et al. [RWLI75] to compute the selected subset S without testing all possible subsets of $\{X, Y\}$. Nevertheless, their algorithm still runs in time exponential in n (Wilfong [Wil91]) in the worst case. However, Wilson and Martinez [WM97] and Wilson [WM00] claim that the average running time of *SNN* is $O(n^3)$. In 1994 Dasarathy [Das94] proposed a complicated algorithm intended to compute a *minimal-size* consistent subset but did not provide a proof of optimality. However, counter-examples to this claim were found by Kuncheva and Bezdek [KB98], Cerverón and Fuertes [CF98] and Zhang and Sun [ZS02]. Wilfong [Wil91] showed in 1991 that the problem of finding the smallest size training-set consistent subset is NP-complete when there are three or more classes. The problem is still open for two classes. Furthermore, he showed that even for only two classes the problem of finding the smallest size consistent *selective* subset (Ritter et al. [RWLI75]) is also NP-complete.

Given the computational difficulty of computing minimum-size training-set consistent subsets, subsequent research has focused on computing small-size training-set consistent subsets. The main challenges regarding the computation of small-size training-set consistent subsets are the scalability for very large datasets as well as for high-dimensional data. An example of this approach is the Fast Condensed Nearest Neighbor rule, which appears to be a promising approach to tackle both problems [Ang07]. Furthermore, since the introduction of the nearest neighbor decision rules more than half a century ago, the Support Vector Machine (SVM) classifier has emerged as a very popular and powerful classifier that rivals the nearest neighbor rules in terms of classification accuracy. Hence one of the main areas of research today is the empirical comparison of the nearest neighbor methods with SVMs in terms of both classification accuracy and computation time, using a wide variety of datasets [GGG+15].

TRAINING-DATA EDITING RULES

Methods have been developed [TBP85] to edit (delete) "redundant" members of $\{X, Y\}$ in order to obtain a subset of $\{X, Y\}$ that implements exactly the same decision boundary as would be obtained using all of $\{X, Y\}$. Such methods depend on the computation of Voronoi diagrams and of other proximity graphs that

are subgraphs of the Delaunay triangulation, such as the Gabriel graph. Furthermore, the fraction of data discarded in such a method is a useful measure of the resulting reliability of the rule. If few vectors are discarded the feature space is relatively empty and more training data are needed. During the past thirty-five years proximity graphs have proven to be very useful both in theory and in practice for solving many of the problems encountered with *NN*-rules. A description of many of these graphs along with related computational geometry problems can be found in [Tou02].

NEAREST-NEIGHBOR SEARCHING

Another important issue in the implementation of nearest-neighbor decision rules, whether editing has or has not been performed, concerns the efficient search for the nearest neighbor of an unknown vector in order to classify it. Various methods exist for computing a nearest neighbor without computing distances to all the candidates. The problem is in general quite difficult when the dimension is high, which it is for most pattern recognition tasks. Simple brute-force search yields $O(dn)$ query time. To improve upon this, one builds a data structure for the points that supports more efficient queries, often at the expense of space for the data structure. For a set of n points in $\mathbb{R}^d$, one could construct a Voronoi diagram for the points of size $O(n^{\lceil d/2 \rceil})$ (Chapter 26), and respond to a query in $O(\log n)$ time. But the exponential space makes this impractical beyond $d \leq 3$. Range searching (Chapter 40) supports structures with linear space and achieving slightly sublinear time. But all constants are exponential in d. This has led to intensive work on approximate nearest-neighbor search, where one seeks a point within $(1+\epsilon)$ of the distance to the true nearest neighbor. An example of an important early milestone along these lines is an algorithm by Arya et al. [AMN+98], which constructs a data structure of size $O(dn)$ that can report approximate nearest neighbors in $O(c \log n)$ time, with $c = O(d(1 + d/\epsilon)^d)$. The algorithm traverses down a balanced box-tree decomposition (BDD) of $O(\log n)$ height and stops when the approximation criterion is satisfied. The query time is logarithmic in n but still the constant is exponential in d. The many advances beyond this and similar algorithms with exponential query time or space requirements are described in Chapters 43 (high dimensions) and 32 (low dimensions).

ESTIMATION OF MISCLASSIFICATION

A very important problem in pattern recognition is the estimation of the performance of a decision rule [McL92]. Many geometric problems occur here also, for which computational geometry offers elegant and efficient solutions. For example, a good method of estimating the performance of the NN-rule is to delete each member of $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ in turn and classify it with the remaining set. Geometrically this problem reduces to computing for a given set of points in d-space the nearest neighbor of each (the ***all-nearest neighbors problem***). Vaidya [Vai89] gives an $O(n \log n)$ time algorithm to solve this problem.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

54.7 PATTERN COMPLEXITY

GLOSSARY

Kolmogorov complexity: The length of the shortest algorithm that generates the pattern.

Papentin complexity: The length of the shortest description of the pattern by means of a specified hierarchy of description languages.

Normalized pairwise variability index (nPVI): The average variation of a set of distances that are obtained from successive adjacent ordered pairs of elements in the pattern.

Sub-symmetries: The contiguous subsets of a pattern that possess reflection symmetries.

Perimetric complexity: The sum of the exterior perimeter of the pattern and the perimeter of the holes in the pattern (interior perimeter), squared, divided by the pattern area, and divided by 4π.

The study of complexity in pattern recognition is one of the more recent developments in the field, and it concerns two rather different problems, the solutions to which require different approaches: the measurement of the complexity of a single pattern, and the complexity of a dataset (training set) of patterns. A single pattern can be one-dimensional, e.g., temporal data such as speech and music, two-dimensional as in character recognition and image processing, or three or higher dimensional. In addition to being of theoretical interest, in practice the complexity of a pattern serves as a feature (shape measurement) of the pattern. For example, in musical rhythm the complexity of a rhythm is akin to the amount of syncopation the rhythm contains. The complexity of a collection of patterns (the training set) is referred to as *data complexity*, where data consist of usually multidimensional feature vectors (points in some space) [HBL06].

The investigation of data complexity via geometrical characteristics of training data is motivated by the desire to determine the existence of independent factors that contribute to the difficulty of pattern classification, and hence guide the selection of an appropriate classification algorithm for a specific application [Can13]. The measures usually considered include measures of overlaps in feature values, measures of the separability of pattern classes, and measures of the shape, density, and interconnectedness of the data manifolds. The emphasis in this research has been on the relative effectiveness of predicting the pattern classification accuracy of several popular decision rules, such as nearest neighbor rules [HBL06] [CRV12]. However, investigation of the relative computational complexity of these geometric measures remains mostly open, apart from a few exceptions such as measures of class separability [Hou93].

Definitions of measures of the complexity of individual patterns vary depending on the definitions of "patterns," which depends in turn on the particular domain that investigates patterns [TT14]. For one-dimensional patterns such as symbol sequences, an appropriate definition for the complexity of an individual pattern

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

(rather than a collection of patterns), is the Kolmogorov complexity [Kol65], which is defined as the length of the shortest algorithm that generates the pattern. Although this measure is theoretically fruitful, it is not computable in practice, and hence there is a need for suitable approximations to the Kolmogorov complexity. One attractive approximation (lower bound) of the Kolmogorov complexity was proposed by Papentin [Pap80] in terms of the shortest description of the pattern computed in a well defined efficiently computable hierarchy of specific description languages.

A measure of one-dimensional pattern complexity that originated in computational linguistics, but has made a significant entry into the study of musical rhythm, is the *normalized pairwise variability index* (*nPVI*) [RAR10]. The *nPVI* is defined as the average variation of a set of distances that are obtained from successive adjacent ordered pairs of elements in the pattern. For a comparison of the *nPVI* to other measures of one-dimensional pattern complexity in the domain of musical rhythm see [Tou13].

Another easily computable measure of the simplicity of one-dimensional patterns is the number of *sub-symmetries* that the pattern contains. A sub-symmetry of a sequence is defined as a contiguous (connected) subsequence of the sequence (pattern) that possesses mirror symmetry. For example, in the pattern BBWWWBB there are four sub-symmetries of length two (BB, WW, WW, and BB), one of length three (WWW), one of length five (BWWWB), and one of length seven (BBWWWBB), for a total of seven. A pattern that contains many sub-symmetries is considered to be simple, and one with few sub-symmetries is complex. This measure was originally shown to predict human judgments of the complexity of visual one-dimensional patterns. However, it was recently discovered that it also predicts human judgments of the complexity of auditory patterns (musical rhythms) [TB13]. These measures can be generalized to apply to two-dimensional patterns. The simplest extension is simply to compute the one-dimensional versions for every row, column and diagonal of the two-dimensional pattern and add up all the counts obtained [TOV15].

A popular measure of pattern complexity with a variety of applications to character recognition is the *perimetric complexity*. For continuous plane shapes the perimetric complexity of a binary pattern with holes is defined as the sum the exterior perimeter of the pattern and the perimeter of the holes (interior perimeter), squared, divided by the pattern area, and divided by 4pi. Several variants for the case of digital images, and for taking blurring into account, are proposed by Watson [Wat12], who also cites many applications of this measure.

54.8 SOURCES AND RELATED MATERIAL

SURVEYS

[BR14]: A survey of traditional and graph theoretical techniques for image segmentation.

[JMF99]: A survey of clustering from the pattern recognition point of view.

[MQ94]: A survey of properties and types of sphere of influence graphs.

[MPW13]: An exhaustive survey of the state-of-the-art in symmetry detection methods for geometric data sets.

[MC15]: An extensive survey of the Hough transform and its variants.

[ABE09]: A survey of algorithms for computing the medial axis.

[Tou91]: A survey of computer vision problems where computational geometry may be applied. This survey references several others; the entire collection is of interest as well.

[Tou00]: A survey on computing nice viewpoints of objects in space.

[Tou14a]: A survey on applications of the relative neighborhood graph to pattern recognition problems.

[Tou14b]: A survey on theoretical results and applications of the sphere of influence graph to pattern recognition problems.

[Tou14c]: A survey on pattern recognition problems, including several distance measures between sets, that may be computed with the *rotating calipers*.

RELATED CHAPTERS

Chapter 1: Finite point configurations
Chapter 10: Geometric graph theory
Chapter 27: Voronoi diagrams and Delaunay triangulations
Chapter 30: Polygons
Chapter 32: Proximity algorithms
Chapter 38: Point location
Chapter 40: Range searching
Chapter 43: Nearest-neighbors in high-dimensional spaces
Chapter 45: Robust geometric computation

REFERENCES

[ABE09] D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes-a state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 109–125, Springer, Berlin, 2009.

[ABL02] A. Apostolico, M.E. Bock, and S. Lonardi. Monotony of surprise and large-scale quest for unusual words. *J. Comput. Biology* 10.3-4: 283-311, 2003.

[ACH91] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:209–206, 1991.

[ADB+10] M.A. Abam, M. de Berg, P. Hachenberger, and A. Zarei. Streaming algorithms for line simplification. *Discrete Comput. Geom.*, 43:497–515, 2010.

[AFL06] G. Aloupis, T. Fevens, S. Langerman, T. Matsui, A. Mesa, Y. Nuñez, D. Rappaport, and G. Toussaint. Algorithms for computing geometric measures of melodic similarity. *Computer Music Journal*, 30:67–76, 2006.

[AG00] H. Alt and L.J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153, Elsevier, Amsterdam, 2000.

[AHM+05] P.K. Agarwal, S. Har-Peled, N.H. Mustafa, and Y. Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42:203–219, 2005.

[AK96] T. Asano and N. Katoh. Variants for the Hough transform for line direction. *Comput. Geom.*, 6:231–252, 1996.

[AMN+98] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM*, 45:891–923, 1998.

[AMW+88] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.*, 3:237–256, 1988.

[Ang07] F. Angiulli. Fast nearest neighbor condensation for large data sets classification. *IEEE Trans. Knowl. Data Eng.*, 19:1450–1464, 2007.

[AT79] S.G. Akl and G.T. Toussaint. Addendum to "An improved algorithm to check for polygon similarity." *Inform. Process. Lett.*, 8:157–158, 1979.

[AW13] F. Aurenhammer and G. Walzl. Three-dimensional straight skeletons from bisector graphs. In *Proc. 5th Internat. Conf. Analytic Number Theory and Spatial Tessellations*, pages 58–59, 2013.

[BC15] D. Belazzougui and F. Cunial. Space-efficient detection of unusual words. In *Proc. 22nd Sympos. String Process. Info. Retrieval*, vol. 9309 of *LNCS*, pages 222–233, Springer, Berlin, 2015.

[BCG98] P. Bose, J. Caron, and K. Ghoudi. Detection of text-line orientation. In *10th Canadian Conf. Comput. Geom.*, Québec, 1998.

[Ber06] P. Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data*, pages 25–71, Springer, Berlin, 2006.

[BGRT99] P. Bose, F. Gómez, P. Ramos, and G.T. Toussaint. Drawing nice projections of objects in space. *J. Visual Commun. Image Rep.*, 10:155–172, 1999.

[BJF90] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *Proc. 10th IEEE Conf. Pattern Recogn.*, pages 820–825, 1990.

[BJW+08] S. Bereg, M. Jiang, W. Wang, B. Yang, and B. Zhu. Simplifying 3D polygonal chains under the discrete Fréchet distance. In *Proc. 8th Latin American Sympos. Theoret. Informatics*, vol. 4957 of *LNCS*, pages 630–641, Springer, Berlin, 2008.

[BK13] A.K. Bajwa, and R. Kaur. Fast lane detection using improved Hough transform. *J. Computing Technologies*, 2:10–13, 2013.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380, MIT Press, Cambridge, 1967.

[BR94] P. Bhattacharya and A. Rosenfeld. Polygons in three dimensions. *J. Visual Communication and Image Representation*, 5:139–147, 1994.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[BR14] B. Basavaprasad and S.H. Ravindra. A survey on traditional and graph theoretical techniques for image segmentation. In *Proc. National Conf. Recent Advances in Information Technology*, pages 38–46, 2014.

[Bra02] P. Brass. Combinatorial geometry problems in pattern recognition. *Discrete Comput. Geom.*, 28:495–510, 2002.

[BT83] B.K. Bhattacharya and G.T. Toussaint. Efficient algorithms for computing the maximum distance between two finite planar sets. *J. Algorithms*, 4:121–136, 1983.

[BT10] J. Bhadury, and C.A. Tovey. An improved implementation and analysis of the Diaz and O'Rourke algorithm for finding the Simpson point of a convex polygon. *Int. J. Computer Math.*, 87:244–259, 2010.

[BWR93] J.B. Burns, R.S. Weiss, and E.M. Riseman. View variation of point-set and line-segment features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:51–68, 1993.

[Can13] J.-R. Cano. Analysis of data complexity measures for classification. *Expert Syst. Appl.*, 40:4820–4831, 2013.

[CC96] W.S. Chan and F.Y.L. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Internat. J. Comput. Geom. Appl*, 6:59–77, 1996.

[CF98] V. Cerverón and A. Fuertes. Parallel random search and Tabu search for the minimum consistent subset selection problem. In *Randomization and Approximation Techniques in Computer Science*, vol. 1518 of *LNCS*, pages 248–259, Springer, Berlin, 1998.

[CRV12] G.D.C. Cavalcanti, I.R. Tsang, and B.A. Vale. Data complexity measures and nearest neighbor classifiers: A practical analysis for meta-learning. In *Proc. 24th IEEE Conf. Tools with Artificial Intelligence*, volume 1, pages 1065–1069, 2012.

[CSW99] F. Chin, J. Snoeyink, and C.A. Wang. Finding the medial axis of a simple polygon in linear time. *Discrete Comput. Geom.*, 21:405–420, 1999.

[CT76] D. Cheriton and R.E. Tarjan. Finding minimum spanning trees. *SIAM J. Comput.*, 5:724–742, 1976.

[CT77] M. Cohen and G.T. Toussaint. On the detection of structures in noisy pictures. *Pattern Recogn.*, 9:95–98, 1977.

[Cul00] T. Culver. Computing the Medial Axis of a Polyhedron Reliably and Efficiently. PhD thesis, University of North Carolina, Chapel Hill, 2000.

[Cyc94] J.M. Cychosz. Efficient binary image thinning using neighborhood maps. In P. Heckbert, editor, *Graphics Gems IV*, pages 465–473. Academic Press, Boston, 1994.

[Das94] B.V. Dasarathy. Minimal consistent set (MCS) identification for optimal nearest neighbor decision system design. *IEEE Trans. Syst. Man Cybern.*, 24:511–517, 1994.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, New York, 1996.

[DM00] G. Das and H. Mannila. Context-based similarity measures for categorical databases. In *Principles of Data Mining and Knowledge Discovery*, pages 201–210, 2000.

[DO94] M. Díaz and J. O'Rourke. Algorithms for computing the center of area of a convex polygon. *Visual Comput.*, 10:432–442, 1994.

[DPR92a] S.J. Dickinson, A. Pentland, and A. Rosenfeld. 3D shape recovery using distributed aspect matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:174–197, 1992.

[DPR92b] S.J. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3D object recognition. *CVGIP: Image Understanding*, 55:130–154, 1992.

[DWT99] S.J. Dickinson, D. Wilkes, and J.K. Tsotsos. A computational model of view degeneracy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:673–689, 1999.

[DZ04] T.K. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 38:179–200, 2004.

[Ead88] P. Eades. Symmetry finding algorithms. In G.T. Toussaint, editor, *Computational Morphology*, pages 41–51, North-Holland, Amsterdam, 1988.

[Ede85] H. Edelsbrunner. Computing the extreme distances between two convex polygons. *J. Algorithms*, 6:213–224, 1985.

[EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13:43–72, 1994.

[ET94] D. Eu and G.T. Toussaint. On approximating polygonal curves in two and three dimensions. *CVGIP: Graph. Models Image Process.*, 56:231–246, 1994.

[FH51] E. Fix and J. Hodges. Discriminatory analysis. Nonparametric discrimination: Consistency properties. Tech. Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[FO12] L.A.F. Fernandes and M.M. Oliveira. A general framework for subspace detection in unordered multidimensional data. *Pattern Recogn.*, 45:3566–3579, 2012.

[GGG+15] M. Gamboni, A. Garg, O. Grishin, S. Man Oh, F. Sowani, A. Spalvieri-Kruse, G.T. Toussaint, and L. Zhang. An empirical comparison of support vector machines versus nearest neighbour methods for machine learning applications. In *Proc. 3rd Conf. Pattern Recogn. Appl. Methods*, vol. 9443 of *LNCS*, pages 110–129, Springer, Berlin, 2015.

[GHS+01] F. Gómez, F. Hurtado, J.A. Sellarès, and G.T. Toussaint. On degeneracies removable by perspective projection. *Internat. J. Math. Alg.*, 2:227–248, 2001.

[Gor96] A.D. Gordon. A survey of constrained classification. *Comput. Stat. Data Anal.*, 21:17–29, 1996.

[GP83] J.E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12:484–507, 1983.

[Gre76] U. Grenander. *Pattern Synthesis*. Springer, New York, 1976.

[GRS00] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Info. Sys.*, 25:345–366, 2000.

[GRT01] F. Gómez, S. Ramaswami, and G.T. Toussaint. On computing general position views of data in three dimensions. *J. Visual Commun. Image Rep.*, 12:387–400, 2001.

[Har68] P.E. Hart. The condensed nearest neighbor rule. *IEEE Trans. Inform. Theory*, 14:515–516, 1968.

[HBL06] T.K. Ho, M. Basu, and M.H.C. Lau. Measures of geometrical complexity in classification problems. In M. Basu and T. Kam Ho, editors, *Data Complexity in Pattern Recognition*, pages 3–23, Springer, London, 2006.

[Hou93] M.E. Houle. Algorithms for weak and wide separation of sets. *Discrete Appl. Math.*, 45:139–159, 1993.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[HS97] F. Hurtado and A.A. Sellarès. Proyecciones perspectivas regulares: Correspondencia por proyeccion perspectiva entre configuraciones planas de puntos. In *Actas VII Encuentros de Geometria Computacional*, pages 57–70, 1997.

[Itt93] D.J. Ittner. Automatic inference of textline orientation. In *Proc. 2nd Sympos. Document Anal. Info. Retrieval*, pages 123–133, 1993.

[JB91] X.-Y. Jiang and H. Bunke. Determination of the symmetries of polyhedra and an application to object recognition. In *Proc. Comput. Geom.: Methods, Algorithms, Appl.*, vol. 553 of *LNCS*, pages 113–121, Springer, Berlin, 1991.

[Jen96] J.R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective*, 2nd edition. Prentice-Hall, Englewood Cliffs, 1996.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31:264–323, 1999.

[JT92] J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80:1502–1517, 1992.

[KB98] L.I. Kuncheva and J.C. Bezdek. Nearest prototype classification: Clustering, genetic algorithms, or random search. *IEEE Trans. Syst. Man Cybern.*, 28:160–164, 1998.

[KCD+04] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A reflective symmetry descriptor for 3D models. *Algorithmica*, 38:201–225, 2004.

[KF87] J. Kender and D. Freudenstein. What is a degenerate view? In *Proc. 10th Joint Conf. Artif. Intell.*, pages 801–804, 1987.

[KK88] T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process.*, 41:43–56, 1988.

[KL02] D. Krznaric and C. Levcopoulos. Optimal algorithms for complete linkage clustering in d dimensions. *Theoret. Comput. Sci.*, 286:1–149, 2002.

[KMN+04] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28:89–112, 2004.

[Kol65] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1:1–7, 1965.

[LAM10] A.J. Law, D.G. Aliaga, and A. Majumder. Projector placement planning for high quality visualizations on real-world colored objects. *IEEE Trans. Vis. Comput. Graphics*, 16:1633–1641, 2010.

[Lea92] V.F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform.* Springer, Berlin, 1992.

[Lee82] D.T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-4:363–369, 1982.

[Liv93] C. Livingston. *Knot Theory.* Math. Assoc. Amer., Washington, 1993.

[LPPD16] T. Lafarge, B. Pateiro-Lopez, A. Possolo, and J.P. Dunkers. R implementation of a polyhedral approximation to a 3D set of points using the α-shape. American Statistical Association, 2014. `https://cran.r-project.org/web/packages/alphashape3d/`.

[LSC97] D.E. Lake, B.M. Sadler, and S.D. Casey. Detecting regularity in minefields using collinearity and a modified Euclidean algorithm. In *Proc. SPIE 3079, Detection and Remediation Technologies for Mines and Minelike Targets II*, 1997.

[LW88] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. 2nd Internat. Conf. on Computer Vision*, pages 238–249, 1988.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Sympos. Math. Stat. Probab.*, pages 281–296, 1967.

[Mar04] D.J. Marchette. *Random Graphs for Statistical Pattern Recognition.* John Wiley & Sons, Hoboken, 2004.

[Mat00] J. Matoušek. On approximate geometric k-clustering. *Discrete Comput. Geom.*, 24:61–84, 2000.

[McL92] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley & Sons, Hoboken, 1992.

[MNV12] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is NP-hard. *Theoret. Comput. Sci.*, 442:13–21, 2012.

[MO88] A. Melkman and J. O'Rourke. On polygonal chain approximation. In G.T. Toussaint, editor, *Computational Morphology*, pages 87–95, North-Holland, Amsterdam, 1988.

[MPW13] N.J. Mitra, M. Pauli, M. Wand, and D. Ceylan. Symmetry in 3D geometry: Extraction and applications. *Computer Graphics Forum*, 22:1–23, 2013.

[MQ94] T.S. Michael and T. Quint. Sphere of influence graphs: A survey. *Congressus Numerantium*, 105:153–160, 1994.

[MSW02] R. Nussinov, M. Shatsky, and H.J. Wolfson. Flexible protein alignment and hinge detection. *Proteins*, 48:242–256, 2002.

[MC15] P. Mukhopadhyay and B.B. Chaudhuri. A survey of Hough transform. *Pattern Recogn.*, 48:993–1010, 2015.

[ÓMa98] D.S. Ó Maidín. A geometrical algorithm for melodic difference. *Computing in Musicology*, 11:65–72, 1998.

[Pap80] F. Papentin. On order and complexity. I. General considerations. *J. Theoret. Biology*, 87:421–456, 1980.

[PB97] N.R. Pal and J. Biswas. Cluster validation using graph theoretic concepts. *Pattern Recogn.*, 30:847–857, 1997.

[PD90] H. Plantinga and C.R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5:137–160, 1990.

[PM99] D. Pelleg and A. Moore. Accelerating exact k-means algorithms with geometric reasoning. In *Knowledge Discovery and Data Mining*, pages 277–281, AAAI Press, New York, 1999.

[PZZ13] B. Peng, L. Zhang, and D. Zhang. A survey of graph theoretical approaches to image segmentation *Pattern Recogn.*, 46:1020–1038, 2013.

[RAR10] M. Raju, E.L. Asu, and J. Ross. Comparison of rhythm in musical scores and performances as measured with the pairwise variability index. *Musicae Scientiae*, 14:51–71, 2010.

[Rei32] K. Reidemeister. *Knotentheorie.* Springer, Berlin, 1932. L.F. Boron, C.O. Christenson and B.A. Smith (English translation), *Knot Theory*, BSC Associates, Moscow, Idaho, USA, 1983.

[Rob93] J.-M. Robert. Maximum distance between two sets of points in E^d. *Pattern Recogn. Lett.*, 14:733–735, 1993.

[RWLI75] G.L. Ritter, H.B. Woodruff, S.R. Lowry, and T.L. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Trans. Inform. Theory*, 21:665–669, 1975.

[SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, vol. 400, pages 525–526, 2000.

[Smi00] M. Smid. Closest point problems in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 877–935, Elsevier, Amsterdam, 2000.

[SNT+92] V. Srinivasan, L.R. Nackman, J.-M. Tang, and S.N. Meshkat. Automatic mesh generation using the symmetric axis transform of polygonal domains. *Proc. IEEE*, 80:1485–1501, 1992.

[SSSS10] R.K. Satzoda, S. Sathyanarayana, T. Srikanthan, and S. Sathyanarayana. Hierarchical additive Hough transform for lane detection. *IEEE Embedded Syst. Lett.*, 2:23–26, 2010.

[Sug84] K. Sugihara. An $n \log n$ algorithm for determining the congruity of polyhedra. *J. Comput. Syst. Sci.*, 29:36–47, 1984.

[TB13] G.T. Toussaint and J.F. Beltran. Subsymmetries predict auditory and visual pattern complexity. *Perception*, 42:1095–1100, 2013.

[TBP85] G.T. Toussaint, B.K. Bhattacharya, and R.S. Poulsen. The application of Voronoi diagrams to nonparametric decision rules. In *Computer Science and Statistics: The Interface*, pages 97–108, 1985.

[TDS+16] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea. 3D skeletons: A state-of-the-art report. *Computer Graphics Forum*, 35:573–597, 2016.

[TM82] G.T. Toussaint and M.A. McAlear. A simple $O(n \log n)$ algorithm for finding the maximum distance between two finite planar sets. *Pattern Recogn. Lett.*, 1:21–24, 1982.

[Tou80] G.T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.

[Tou84] G.T. Toussaint. An optimal algorithm for computing the minimum vertex distance between two crossing convex polygons. *Computing*, 32:357–364, 1984.

[Tou91] G.T. Toussaint. Computational geometry and computer vision. In B. Melter, A. Rosenfeld, and P. Bhattacharya, editors, *Vision Geometry*, pages 213–224, AMS, Providence, 1991.

[Tou00] G.T. Toussaint. The complexity of computing nice viewpoints of objects in space. In *Proc. Vision Geometry IX, SPIE Internat. Sympos. Optical Sci. Tech.*, pages 1–11, 2000.

[Tou02] G.T. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *Proc. 34th Sympos. Comput. Statist. (Geoscience and Remote Sensing)*, Montreal, 2002.

[Tou05] G.T. Toussaint. Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining. *Internat. J. Comput. Geom. Appl.*, 15:101–150, 2005.

[Tou13] G.T. Toussaint. The pairwise variability index as a measure of rhythm complexity. *Analytical Approaches to World Music*, 2:1–42, 2013.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[Tou14a] G.T. Toussaint. Applications of the relative neighborhood graph. *Int. J. Advances Comp. Sci. Appl.*, 4:77–85, 2014.

[Tou14b] G.T. Toussaint. The sphere of influence graph: Theory and applications. *Int. J. Inform. Tech. Comp. Sci.*, 14:37–42, 2014.

[Tou14c] G.T. Toussaint. Applications of the rotating calipers to geometric problems in two and three dimensions. *Int. J. Digital Information and Wireless Communications*, 4:108–122, 2014.

[TOV15] G.T. Toussaint, N. Onea, and Q. Vuong. Measuring the complexity of two-dimensional patterns: Sub-symmetries versus Papentin complexity. In *Proc. 14th IAPR Conf. Machine Vision Appl.*, pages 80–83, 2015.

[TT14] E.R. Toussaint and G.T. Toussaint. What is a pattern? *Proc. Bridges: Mathematics, Music, Art, Architecture, Culture*, pages 293–300, 2014.

[Vai89] P.M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete Comput. Geom.*, 4:101–115, 1989.

[Vat11] A. Vattani. k-means requires exponentially many iterations even in the plane. *Discrete Comput. Geom.*, 45: 596–616, 2011.

[Wat12] A.B. Watson. Perimetric complexity of binary digital images. *The Mathematica Journal*, 14:1–40, 2012.

[Wil91] G. Wilfong. Nearest neighbor problems. In *Proc. 7th Sympos. Comput. Geom.*, pages 224–233, ACM Press, 1991.

[WM97] D.R. Wilson and T.R. Martinez. Instance pruning techniques. In D. Fisher, editor, *Proc. 14th Internat. Conf. Machine Learning*, pages 404–411, Morgan Kaufmann Publishers, San Francisco, 1997.

[WM00] D.R. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Mach. Learning*, 38:257–286, 2000.

[WWV85] J.D. Wolter, T. Woo, and R.A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *Visual Comput.*, 1:37–48, 1985.

[Yap90] C.K. Yap. Symbolic treatment of geometric degeneracies. *J. Symb. Comput.*, 10:349–370, 1990.

[YD15] Q. Ye and D.S. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37:1480–1500, 2015.

[YL04] J. Yang and I. Lee. Cluster validity through graph-based boundary analysis. In *Proc. Int. Conf. Information and Knowledge Engineering*, pages 204–210, 2004.

[YR91] C. Yao and J.G. Rokne. A straightforward algorithm for computing the medial axis of a simple polygon. *Internat. J. Comput. Math.*, 39:51–60, 1991.

[ZA15] N.M. Zaitoun and M.J. Aqel. Survey on image segmentation techniques. *Procedia Comput. Sci.*, 65:797–806, 2015.

[ZS02] H. Zhang and G. Sun. Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recogn.*, 35:1481–1490, 2002.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.