Similarity Search in Biological and Engineering Databases

Hans-Peter Kriegel, Martin Pfeifle, Stefan Schönauer
Institute for Informatics
University of Munich
{kriegel, pfeifle, schoenauer}@dbs.ifi.lmu.de

Abstract

Similarity search is an important task in a wide range of scientific database applications. Besides being used directly, it is also used as a basic operation for many data mining algorithms. Example applications which are presented in this article are functional classification of proteins in biological databases and the similarity search of CAD parts in engineering environments. It turns out that the general concept of set-based similarity measures can be successfully applied to both of these diverse application domains.

1 Introduction

In the last ten years, an increasing number of database applications has emerged for which efficient and effective support for similarity search is substantial. The importance of similarity search grows in application areas such as multimedia, medical imaging, molecular biology, computer aided engineering, marketing and purchasing assistance, etc. [Jag91, MG93, Fal+94, ALSS95, BKK97, Kei99]. Particularly, the task of finding similar shapes in 2D and 3D becomes more and more important. Examples for new applications that require the retrieval of similar 3D objects include databases for molecular biology, medical imaging and computer aided design.

In recent years, considerable work on similarity search in database systems has been published. Many of the previous approaches, however, deal with 1D or 2D data, such as time series, digital images or polygonal data. In the following, we will shortly list different approaches to establish similarity measures known from literature. We provide a classification of the techniques into feature-based models and direct geometric models.

Feature-Based Similarity. A widely used class of similarity models is based on the paradigm of feature vectors. The basic idea is as follows: Using a feature transform, the objects are mapped onto a feature vector in an appropriate multidimensional feature space. The similarity of two objects is then defined as the proximity of their feature vectors in the feature space: The closer their feature vectors are located, the more similar two objects are considered.

The paradigm of feature-based similarity has been successfully applied to the retrieval of similar spatial objects. Examples include structural features of 2D contours [MG93], angular profiles of polygons [BMH92], rectangular covers of shapes [Jag91], algebraic moment invariants [Fal+94], 2D section coding [BKK97], and 3D shape histograms for biomolecular objects [AKKS99]. Non-geometric applications include similarity search on time series [ALSS95], and on color histograms in image databases [Fal+94], among several others.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

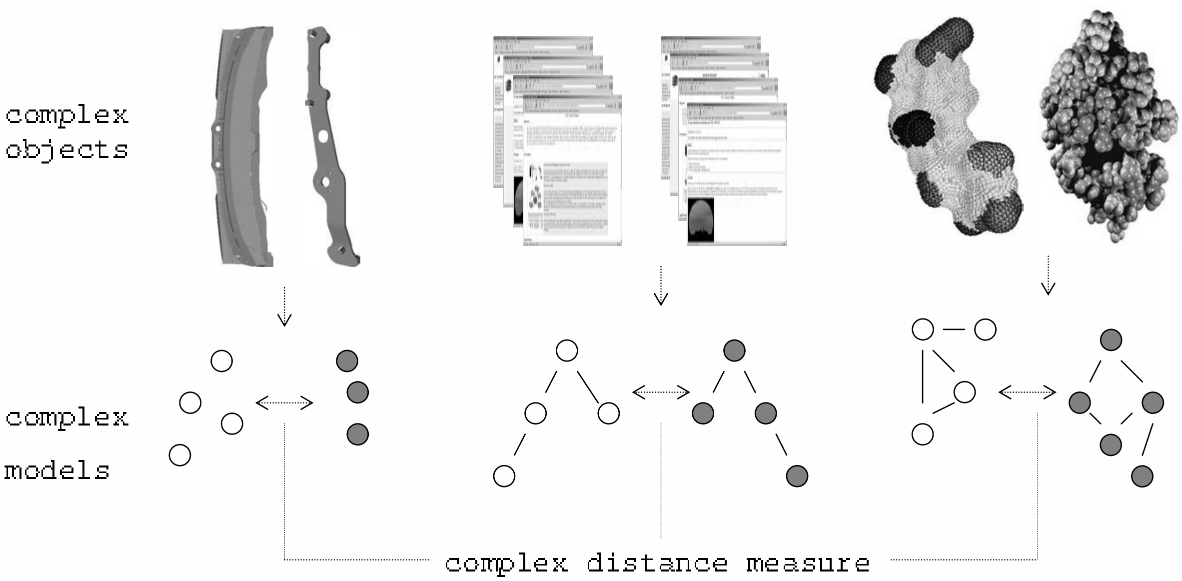

Figure 1: Examples of Complex Objects.

Geometry-Based Similarity. A class of models that is to be distinguished from the feature-based techniques are the similarity models that are defined by directly using the geometry of the objects. Two objects are considered similar if they minimize a distance criterion that is purely defined by the geometry of the objects. Examples include the similarity retrieval of mechanical parts [SKSH89], the difference volume approach [Kei99], and the approximation-based similarity model for 3D surface segments [KSS97].

Model-Based Similarity. In this paper, we propose to model complex real world objects by complex models (cf. Figure 1). As the development of conventional database systems in the recent two decades has shown, the use of more sophisticated ways to model data can enhance both the effectiveness and efficiency for applications using large amounts of data. In this paper, we will use point sets to describe CAD objects and graphs are used to represent protein molecules. The distance measure is in both cases based on a minimum weight perfect matching. Furthermore, we will sketch suitable filters for this rather complex distance function to speed up similarity query processing.

The remainder of the paper is organized as follows. In Section 2, we present a similarity model for voxelized CAD objects which are represented by sets of feature vectors. In Section 3, we present a similarity model for proteins which are modeled as graphs. The distance function used for measuring the similarity between two graphs, is based on the distance function used for measuring the similarity between two CAD objects. We close this paper in Section 4 with a short summary and a few remarks on future work.

2 Engineering Databases

The development, design, manufacturing and maintenance of modern engineering products is a very expensive and complex task. Effective similarity models are required for two- and three-dimensional CAD applications to cope with rapidly growing amounts of data. Shorter product cycles and a greater diversity of models are becoming decisive competitive factors in the hard-fought automobile and aircraft markets. These demands can only be met if the engineers have an overview of already existing CAD parts. In Section 2.2, we present a similarity model for complex 3D CAD data, which helps to find and group similar parts. This model is based on the cover sequence model which is introduced in the following section.

2.1 The Cover Sequence Model

In this section, we adapt a known model [Jag91, JB91] to voxelized 3D data, the so-called *cover sequence model*. The basic idea of this model is to find large clusters of grid primitives, called *covers*, which approximate the object in a best possible way [JB91].

The quality of such a cover sequence S_k is measured by the symmetric volume difference Err_k between the object O and the sequence S_k. Formally, let the covers be drawn from the set $\mathcal{C}$ of all possible rectangular covers. Then each unit i of the cover sequence comprises a pair $(C_i \in \mathcal{C}, \sigma_i \in \{+,-\})$, where "+" represents set union and "−" represents set difference. The sequence after k units is:

$$S_k = (((C_0 \sigma_1 C_1) \sigma_2 C_2) \ldots \sigma_k C_k),$$

where C_0 is an initially empty cover at the origin.

The symmetric volume difference after k units is:

$$Err_k = \ | O \text{ XOR } S_k \,| \text{ , where } O \text{ is the approximated object.}$$

In [Jag91], Jagadish sketches how a 3D cover sequence $S_k = (((C_0\sigma_1 C_1)\sigma_2 C_2) \ldots \sigma_k C_k)$ of an object O, can be transformed into a $6 \cdot k$-dimensional feature vector. Thereby, each cover C_{i+1} with $0 \leq i \leq k-1$ is mapped onto 6 values in the feature vector f_o in the following way:

$$\begin{aligned}
f_o^{6i+1} &= x\text{-position of } C_{i+1} \\
f_o^{6i+2} &= y\text{-position of } C_{i+1} \\
f_o^{6i+3} &= z\text{-position of } C_{i+1} \\
f_o^{6i+4} &= x\text{-extension of } C_{i+1} \\
f_o^{6i+5} &= y\text{-extension of } C_{i+1} \\
f_o^{6i+6} &= z\text{-extension of } C_{i+1}
\end{aligned}$$

2.2 Using Sets of Feature Vectors for Similarity Queries

As proposed in [Jag91] a data object is now represented as a feature vector. For similarity queries this method yields a major problem. Always comparing the two covers having the same ranking according to the symmetric volume difference, does not make sense in all cases. Two objects can be considered very different, because of the order of their covers, although they are very similar by intuition. The reason for this effect is that the order of the covers does not guarantee that the most similar covers due to size and position will be stored in the same dimensions. Especially for objects generating two or more covers having almost the same volume, the intuitive notion of similarity can be seriously disturbed. Thus, the possibility to match the covers of two compared objects using more degrees of freedom, might offer a better similarity measure.

The representation of extracted features as a set of vectors, i.e. each cover is represented by a 6-dimensional feature vector, is a generalization of the use of just one large feature vector. It is always possible to restrict the model to a feature space, in which a data object will be completely represented by just one feature vector. But in some applications the possibilities of vector set representation allow us to model the dependencies between the extracted features more precisely. In our application the vector set representation is able to avoid the problems that occur by storing a set of covers according to a strict order. Therefore, it is possible to compare two objects more intuitively, causing a relatively small rise of calculation cost compared to the cost of distance calculations in the one-vector model.

2.2.1 Distance Measures on Vector Sets

A distance measure on vector sets that demonstrates to be suitable for defining similarity in our application is based on the *minimum weight perfect matching* of sets. This well known graph problem can be applied here, by building a complete bipartite graph $G = (S_1 \cup S_2, E)$ between the vector sets S_1 and S_2. The weight of each edge $(x, y) \in E$ with $x \in S_1$ and $y \in S_2$ in this graph G is defined by their distance $dist(x, y)$. A perfect matching is a subset $M \subseteq E$ that connects each $x \in S_1$ to exactly one $y \in S_2$ and vice versa. A minimum weight perfect matching is a matching with a minimum sum of weights of its edges. Since a perfect match can only be found for sets of equal cardinality, it is necessary to introduce weights for unmatched nodes when defining a distance measure.

Definition 1: (enumeration of a set)
Let S be any finite set of arbitrary elements. Then π is a mapping that assigns $s \in S$ a unique number $i \in \{1, .., |S|\}$. This is written as $\pi(S) = (s_1, .., s_{|S|})$. The set of all possible enumerations of S is named $\Pi(S)$.

Definition 2: (minimal matching distance)
Let O be the domain of the objects and X be a set with $|X| \leq k$ and $X \subseteq 2^V$ where $V \subset I\!R^d$. Furthermore, let $F : O \rightarrow X$ be a mapping of the objects into X, and *dist* : $I\!R^d \times I\!R^d \rightarrow I\!R$ a distance function between two d-dimensional feature vectors. We assume w.l.o.g. $|F(Obj_1)| = m \geq n = |F(Obj_2)|$, $F(Obj_1) = \{x_1, .., x_m\}$ and $F(Obj_2) = \{y_1, .., y_n\}$. Then $dist_{mm}^{w,dist}$: $O \times O \rightarrow I\!R$ is defined as follows:

$$dist_{mm}^{w,dist}(Obj_1, Obj_2) = \min_{\pi \in \Pi(F(Obj_1))} \left(\sum_{i=1}^{n} dist(x_{\pi(i)}, y_i) + \sum_{l=n+1}^{m} w(x_{\pi(l)}) \right)$$

where $w : I\!R^d \rightarrow I\!R^+$ is a weight function for the unmatched elements.

The weight function w provides the penalty given to every unassigned element of the set having larger cardinality. Let us note that *minimum matching distance* is a specialization of *netflow distance* which is introduced in [RB00]. In [RB00] it is proven that netflow distance is a metric and that it is computable in polynomial time. Though it was shown that the netflow distance can be calculated in polynomial time, it is not obvious how to achieve it. Since we are only interested in the minimum matching distance, it is enough to calculate a minimum weight perfect matching. Therefore, we apply the method proposed by Kuhn [Kuh55] and Munkres [Mun57]. The method is based on the successive augmentation of an alternating path between both sets. Since it is guaranteed that this path can be expanded by one further match within each step taking $O(k^2)$ time and since there is a maximum of k steps, the all over complexity of a distance calculation using the method of Kuhn and Munkres is $O(k^3)$ in the worst case. Let us note that for larger numbers of k this is far better than the previously mentioned method on $k!$ many permutations.

For more details about the presented similarity model based on sets of feature vectors, we refer the interested reader to [BKK+03]. In [BKK+03], there is also a $centroid$ filter presented which helps to accelerate similarity queries on vector set represented objects. The basic idea is that for each vector set we compute its centroid. The difference between the two centroids of the two point sets times the cardinality of the point sets forms a lower-bound for the minimal matching distance.

3 Biological Databases

Some of the concepts developed for engineering databases can also be used in biological databases. Exemplarily, we describe the application in the field of protein function prediction.

3.1 Analysis and Prediction of Protein Function

For the analysis and prediction of the interaction of proteins, the molecular surface is of particular interest. The 3D geometry of molecules is a highly selective criterion for the functional interaction of proteins. Advanced experimental methods for structure determination and the application of algorithms to predict protein folds result in tens of thousands of reliably known molecular conformations. Faced with these increasing data volumes, the introduction of database techniques to manage the data becomes more and more vital. One important task in this context is the functional classification of proteins based on homology search. This task can be accomplished using knn-classifiers for the functionally active 3D-surface of protein molecules.

To represent such complex objects as biologically active molecules, rich models integrating geometrical, structural and biochemical information are needed. Therefore, we employ the concept of feature graphs as a model for protein surfaces, which are a type of attributed graphs. They not only allow for the integration of biochemical information, but also offer the possibility to consider molecular flexibility, a key feature in this application domain. This is the basis for an effective functional classification of protein molecules.

3.2 Similarity of Graphs

As graphs are a very general object model, graph similarity has been studied in many fields. Similarity measures for graphs have been used in systems for shape retrieval [HCH99], object recognition [KKV90] or face recognition [WFKM97].

A very common similarity measure for graphs is the edit distance. It uses the same principle as the well known edit distance for strings [Lev66, WF74]. The idea is to determine the minimal number of insertion and deletions of vertices and edges to make the compared graphs isomorphic. In [SF83], Sanfeliu and Fu extended this principle to attributed graphs, by introducing vertex relabeling as a third basic operation beside insertions and deletions. Unfortunately, the edit distance is a very time-complex measure. Zhang, Statman and Shasha proved in [ZSS92] that the edit distance is MAX–SNP–hard even for unordered labeled trees.

In the field of image retrieval, similarity of attributed graphs is sometimes described as an assignment problem [Pet02], where the similarity distance between two graphs is defined as the minimal cost for mapping the vertices of one graph to those of another graph. With an appropriate cost function for the assignment of vertices, this measure takes the vertex attributes into account and can be evaluated in polynomial time. This assignment measure, which we will call vertex matching distance in the following, obviously completely ignores the structure of the graphs, i.e. they are just treated as sets of vertices.

As we just described, all the known similarity measures for attributed graphs have certain drawbacks. Starting from the edit distance and the vertex matching distance we proposed a new method to measure the similarity of attributed graphs. This method solves the problems mentioned above and is useful in the context of large databases of structured objects. For our similarity measure, called the edge matching distance, we rely on the principle of graph matching, just like in the case of the vertex matching distance. But instead of matching the vertices of two graphs, we propose a cost function for the matching of edges and then derive a minimal weight maximal matching between the edge sets of two graphs. This way not only the attribute distribution, but also the structural relationships of the vertices are taken into account.

Definition 3 (edge matching distance): Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two attributed graphs. Without loss of generality, we assume that $|E_1| \geq |E_2|$. The complete bipartite graph $G_{em}(V_{em} = E1 \cup E_2 \cup \Delta, E_1 \times (E_2 \cup \Delta))$, where Δ represents an empty dummy edge, is called the edge matching graph of G_1 and G_2. An edge matching between G_1 and G_2 is defined as a maximal matching in G_{em}. Let there be a non-negative metric cost function $c : E_1 \times (E_2 \cup \Delta) \rightarrow I\!R_0^+$. The edge matching distance between G_1 and G_2, denoted by $d_{match}(G_1, G_2)$, is defined as the cost of the minimum-weight edge matching between G_1 and G_2 with respect to the cost function c.

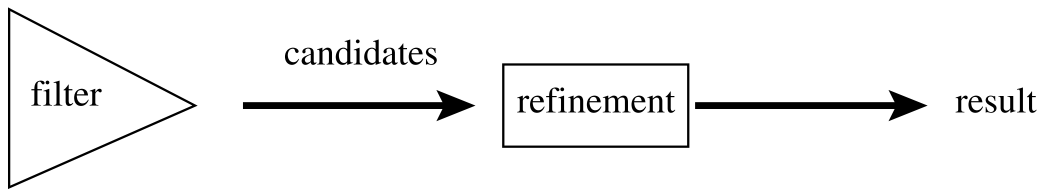

Figure 2: A filter-refinement architecture.

This measure is similar to the matching distance of definition 2. Especially, the algorithm of Kuhn [Kuh55] and Munkres[Mun57] can be used to determine the minimum-weight matching, which results in a time-complexity of $O(n^3)$ where n denotes the number of edges, i.e. $n = |E_1|$.

The edge matching distance has several properties, which are important for the application of the measure to similarity search and to classification systems. One is the fact that the edge matching distance is a metric. This allows the application of several methods to speed up query processing. Another such property is the polynomial time-complexity of the measure as opposed to the exponential time-complexity of the edit distance. The polynomial time-complexity is the precondition for the application of the measure in large databases, where the measure has to be calculated repeatedly even when a single query is processed.

As experiments show, the time-complexity of the edge matching distance is still too high for use in a classification system. Therefore we developed techniques for efficient query processing with the edge matching distance, based on the concept of multi-step query processing.

Query processing in a multi-step query processing architecture, as depicted in figure 2, is performed in two or more steps, where the first steps are filter steps that return a number of candidate objects from the database. For those candidate objects, the exact similarity distance is determined in the refinement step and the objects fulfilling the query predicate are reported. To reduce the overall search time, the filter steps have to be cheap to perform and a substantial part of the database objects has to be filtered out.

Additionally, the completeness of the filter step is essential, i.e. there must be no false drops during the filter steps. Available similarity search algorithms guarantee completeness if the distance function in the filter step fulfills the lower-bounding property. This means that the filter distance between two objects must always be less than or equal to their exact similarity distance.

Using a multi-step query processing architecture requires efficient algorithms which actually make use of the filter step. Agrawal, Faloutsos and Swami proposed such an algorithm for range search [AFS93]. In [SK98] and [KSF+98] multi-step algorithms for k-nearest-neighbor search were presented, which are optimal in the number of exact distance calculations necessary during query processing. Therefore, we use the latter algorithms in our classification system.

To employ a filter-refinement architecture we need filters for the edge matching distance, which cover the structural as well as the attribute properties of the graphs in order to be effective. A way to derive a filter for a similarity measure is to approximate the database objects and then determine the similarity of those approximations. As an approximation for the structure of a graph we use the size of that graph, i.e. the number of edges in the graph. A lower bound for the edge matching distance between two graphs can be derived from the difference between their sizes. Our filters for the attribute part of graphs are based on the observation that the difference between the attribute distributions of two graphs influences their edge matching distance. Obviously, it is too complex to determine the exact difference of the attribute distributions of two graphs in order to use this as a filter and, therefore, an approximation of those distributions is needed. We propose a filter for the attribute part of graphs, which exploits the fact that $\forall x, y \in I\!R : |x - y| \geq ||x| - |y||$. For attributes which are associated with edges, we add all the absolute values for an attribute in a graph. For two graphs G_1 and G_2 with equal size, the difference between those sums is the minimum total difference between G_1 and G_2 for the respective attribute. Weighted appropriately according to the cost function that is used, this is a lower bound for the edge matching distance. For graphs of different size, this is no longer true, as an edge causing the attribute difference could also be assigned to an empty edge. Therefore, the difference in size of the graphs multiplied with the maximum cost for this attribute has to be subtracted from the previously computed value, in order to be lower bounding in all cases.

When considering attributes that are associated with vertices in the graphs, we have to take into account that during the distance calculation a vertex v is compared with several vertices of the second graph, namely exactly $degree(v)$ many vertices. To take care of this effect, the absolute attribute value for a vertex attribute has to be multiplied with the degree of the vertex, which carries this attribute value, before the attribute values are added in the same manner as for edge attributes. Obviously, the appropriately weighted size difference has to be subtracted in order to achieve a lower bounding filter value for a node attribute.

With the above methods it is ensured that the sum of the structural filter distance plus all attribute filter distances is still a lower bound for the edge matching distance between two graphs. Furthermore, it is possible to precompute the structural and all attribute filter values and store them in a single vector. This supports efficient filtering during query processing.

The edge matching distance and the filtering methods are more thoroughly described in [KS03].

First experiments on the effectiveness and efficiency of our classification approach already show encouraging results. This underlines the applicability of general concepts developed for engineering databases in the area of biological databases.

4 Conclusions

Similarity search is an important task in a wide range of scientific database applications. Besides being used on its own, it is also a basic operation for many data mining applications. In this paper, we presented two such data mining applications from the areas of engineering and biology. With the *minimal matching distance* and the *edge matching distance* we demonstrated that the concept of a similarity measure between sets can be successfully applied in two very diverse application areas. Additionally, the presented similarity measure can be efficiently handled even in large databases using multi-step query processing architectures.

In our future work, we would like to extend the presented concepts to other application domains as well to new problems in engineering and bioinformatics.

References

[AFS93] Agrawal, R. and Faloutsos, C. and Swami, A. N.: *Efficient similarity search in sequence databases*. Proc. 4th Int. Conf. of Foundations of Data Organization and Algorithms (FODO), Chicago, Illinois, pp. 69–84, 1993.

[AKKS99] Ankerst, M. and Kastenmüller, G. and Kriegel, H.-P. and Seidl T.: *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*. Proc. 6th Int. Symposium on Large Spatial Databases (SSD'99), Hong Kong, China, LNCS 1651, pp. 207–226, 1999.

[ALSS95] Agrawal, R. and Lin, K.-I. and Sawhney, H.S. and Shim, K.: *Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases*. Proc. 21th Int. Conf. on Very Large Databases (VLDB'95), pp. 490–501, 1995.

[BKK97] Berchtold, S. and Keim, D.A. and Kriegel, H.-P.: *Using Extended Feature Objects for Partial Similarity Retrieval*. VLDB Journal, 6(4):333–348, 1997.

[BKK+03] Brecheisen S. and Kriegel H.-P. and Kröger P. and Pfeifle M. and Schubert M.: *Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects*. Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03), San Diego, CA, pp. 587-598, 2003.

[BMH92] Badel, A. and Mornon, J.P. and Hazout, S.: *Searching for Geometric Molecular Shape Complementarity using Bidimensional Surface Profiles*. Journal of Molecular Graphics, 10: 205–211,1992.

[Fal+94] Faloutsos, C. and Barber, R. and Flickner, M. and Hafner, J. and others: *Efficient and Effective Querying by Image Content*. Journal of Intelligent Information Systems, 3:231–262, 1994.

[HCH99] Huet, B. and Cross, A. and Hancock, E.: *Shape Retrieval by Inexact Graph Matching*. Proc. IEEE International Conference on Multimedia Computing Systems, pp. 40–44, 1999.

[Jag91] Jagadish, H.V.: *A Retrieval Technique for Similar Shapes*. Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'91), pp.208–217, 1991.

[JB91] Jagadish, H. V. and Bruckstein, A. M.: *On sequential shape descriptions*. Pattern Recocnition, 1991.

[Kei99] Keim, D.A.: *Efficient Geometry-based Similarity Search of 3D Spatial Databases*. Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), Philadelphia, PA, pp. 419–430, 1999.

[KKV90] Kubicka, E. and Kubicki, G. and Vakalis, I.: *Using Graph Distance in Object Recognition*. Proc. ACM Computer Science Conference, pp. 43–48, 1990.

[KS03] Kriegel, H.-P. and Schönauer, S.: *Similarity Search in Structured Data*. Proc. 5th Int. Conf. DaWaK 2003, Prague, Czech Republic, LNCS 2992, pp. 676–693, 2004.

[KSF+98] Korn,K. and Sidiropoulos, N. and Faloutsos, C. and Siegel, E. and Protopapas, Z.: *Fast and effective retrieval of medical tumor shapes*. IEEE Transactions on Knowledge and Data Engineering, 10(6), pp. 889–904, 1998.

[KSS97] Kriegel, H.-P. and Schmidt, T. and Seidl, T.: *3D Similarity Search by Shape Approximation*. Proc. 5th Int. Symposium on Large Spatial Databases (SSD'97), Berlin, Germany, volume 1262 of Lecture Notes in Computer Science (LNCS), pp. 11–28, 1997.

[Kuh55] Kuhn, H.W.: *The Hungarian method for the assignment problem*. Naval Research Logistics Quarterly, 2:83-97, 1995.

[Lev66] Levenshtein, V.: *Binary codes capable of correcting deletions, insertions and reversals*. Soviet Physics-Doklady, 10, pp. 707–710, 1966.

[MG93] Mehrotra, R. and Gary, J.E.: *Feature-Based Retrieval of Similar Shapes*. Proc. 9th Int. Conf. on Data Engineering, Vienna, Austria, pp. 108–115, 1993.

[Mun57] Munkres, J.: *Algorithms for the assignment and transportation problems*. Journal of the SIAM, 6:32-38, 1957.

[Pet02] Petrakis, E.: *Design and evaluation of spatial similarity approaches for image retrieval*. Image and Vision Computing, 20(1), pp. 59–76, 2002.

[RB00] Ramon, J. and Bruynooghe, M.: *A polynomial time computable metric between point sets*. Acta Informatica, 37: 765-780, 2001.

[SF83] Sanfeliu, A. and Fu, K.-S.: *A distance measure between attributed relational graphs for pattern recognition*. IEEE Transactions on Systems, Man and Cybernetics, 13(3), pp. 353–362, 1983.

[SK98] Seidl, T. and Kriegel, H.-P.: *Optimal multi-step k-nearest neighbor search*. Proc. 17th ACM SIGMOD, Seattle, WA, pp. 154–165, 1998.

[SKSH89] Schneider, R. and Kriegel, H.-P. and Seeger, B. and Heep, S.: *Geometry-based Similarity Retrieval of Rotational Parts*. Proc. Int. Conf. on Data and Knowledge Systems for Manufacturing and Engineering, Gaithersburg, ML, pp. 150–160, 1989.

[WF74] Wagner, R. A. and Fisher, M. J.: *The String-to-String Correction Problem*. Journal of the ACM, 21(1), pp. 168–173, 1974.

[WFKM97] Wiskott, L. and Fellous, J.-M. and Krüger, N. and von der Malsburg, C.: *Face Recognition by Elastic Bunch Graph Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), pp. 775–779, 1997.

[ZSS92] Zhang, K. and Statman, R. and Shasha, D.: *On the editing distance between unordered labeled trees*. Information Processing Letters, 42, pp. 133–139, 1992.