

A Descriptive Tolerance Nearness Measure for Performing Graph Comparison

Christopher J. Henry* C

Department of applied computer science

University of Winnipeg

Portage avenue R3B 2E9, Manitoba, Canada

ch.henry@uwinnipeg.ca

Syed Aqeel Awais

Department of applied computer science

University of Winnipeg

Manitoba, Canada

Abstract. This article proposes the tolerance nearness measure (TNM) as a computationally reduced alternative to the graph edit distance (GED) for performing graph comparisons. The TNM is defined within the context of near set theory, where the central idea is that determining similarity between sets of disjoint objects is at once intuitive and practically applicable. The TNM between two graphs is produced using the Bron-Kerbosh maximal clique enumeration algorithm. The result is that the TNM approach is less computationally complex than the bipartite-based GED algorithm. The contribution of this paper is the application of TNM to the problem of quantifying the similarity of disjoint graphs and that the maximal clique enumeration-based TNM produces comparable results to the GED when applied to the problem of content-based image processing, which becomes important as the number of nodes in a graph increases.

Keywords: Graph matching, Tolerance nearness measure, Graph edit distance, Near sets, Maximal clique enumeration

Address for correspondence: University Of Winnipeg, Department of Applied Computer Science, Winnipeg, R3B 2E9, Canada

*The authors extend their profound thanks to Francesc Serratos and Xavier Cortés for providing a Matlab implementation of their GED algorithm. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant 418413.

^CCorresponding author

1. Introduction

This article proposes the tolerance nearness measure (TNM) as a computationally reduced alternative to the graph edit distance (GED) for performing graph comparisons. The graph comparison problem consists of quantifying the similarity or dissimilarity of graphs [1] and is required by many applications [2]. The TNM is example of a graph comparison measure, while the GED is an example of a graph matching algorithm. The chief difference between graph comparison and graph matching algorithms is that a graph matching algorithm maps similar substructures between the two graphs under comparison to produce a real value reflecting the degree these substructures align. Graph comparison methods are a generalization of graph matching approaches where the only requirement is assigning a real value indicating the degree of similarity between the two graphs, *i.e.* the likeness between the two graphs does not necessarily depend on matching substructures. In fact, the TNM approach proposed here is based on finding maximal cliques distributed across the two graphs in question. Further, it is important to note that the graph comparison problem is commonly solved by graph matching algorithms¹ [1]. This observation provides a basis for the comparison presented in this paper. In other words, both the GED and TNM make judgements regarding nearness and similarity of attributed graphs.

Many distance measures have been proposed to solve the graph comparison problem. The TNM is defined within the context of near set theory, where the central idea is that determining similarity between sets of disjoint objects is at once intuitive and practically applicable. More importantly, near set theory provides a mathematically-rigorous, theoretical framework for making decisions regarding nearness or apartness. The TNM grew out of work on practical applications of descriptive near set theory, and is calculated by using a maximal clique enumeration (MCE) algorithm. To date, the TNM has only been used to quantify the similarity of disjoint sets of objects with associated feature vectors. Here, the TNM is used to quantify the similarity of graphs as a solution to the graph comparison problem. In this regard, the TNM is applied to graphs in two ways: in its original form to assess the similarity between the vertices of two graphs (*i.e.* two disjoint sets of objects) based on their attributes; and to assess the similarity of two graphs using both vertex and edge information. The presented approach is contrasted with the well-known GED matching algorithm through application to the problem of content-based image retrieval (CBIR) on the Tarragona Rotation Zoom [3, 4], SIMPLiCity [5] and the Palmprint datasets [6]. The GED algorithm was selected due to its popularity and benchmark status within the field of graph comparisons [7]. The contribution of this paper is the application of TNM to the problem of quantifying the similarity of disjoint graphs – in the context of descriptive near set theory – and that the MCE-based TNM produces comparable results to the GED when applied to the problem of content-based image retrieval, which becomes important as the number of nodes in a graph increases. This article is organized as follows. Section 2 gives background on the GED and TNM, Section 3 describes the graph comparison measures and provides a detailed explanation of the GED and TNM. Section 4 presents the results along with discussions, and, finally, Section 5 is the conclusion of the article.

¹The graph comparison problem is also referred to as *graph similarity*, *graph distance*, or *graph measure* in the literature. Examples using this terminology are cited in the last paragraph of Section 2.

2. Background

A comprehensive survey of most graph matching algorithms proposed since the late seventies, together with pattern recognition and machine vision applications, is presented in [8]. With respect to applications, there are numerous examples of solving practical problems using graph similarity measures. Examples include structural pattern recognition [9, 10], graph classification [11], character recognition [12], artificial intelligence [13, 14], chemoinformatics [15], document analysis [16], and bioinformatics [17]. Other interesting examples include shape analysis [18] and 3D segmentation [19].

In this work, the focus is on the popular GED [20, 21, 20, 22], which is considered one of the most well known graph matching approaches since it is generic and easily interpretable [22, 10, 7] (see, [23, 1] for a extensive discussions on the GED). The GED is an error-tolerant, inexact graph similarity measure. Accordingly, it can be applied to graphs with variable number of nodes, such as the graphs formed by the content-based image retrieval (CBIR) application presented in this paper.

The GED was first used for string matching, introduced as the *edit distance*, where it was used to find the minimum cost for all sequences of edit operations required to transform one string to other [24, 25]. This was followed by the adaptation of the edit distance for use in comparing trees [26]. In this case, it was modified to find the minimum amount of edit operations to convert one tree into the other, where each operation was associated with specific cost. Finally, the GED appears in its present form in 1983 [27, 21, 20]. Since then, it has been used in many applications: fingerprint classification [28], graph clustering problems [29], chemical graph recognition problem [30], molecular electrostatic potentials comparison [31] and diatom identification [32]. Some of the more recent examples include: database management [33], biological network alignment [34] and handwriting recognition [12, 35]. In this paper, the GED and the TNM approach are both applied to the problem of CBIR.

While the GED is well established, the TNM was recently introduced in 2010 [36] within the context of descriptive near sets theory². In general, the TNM follows from a rich and rigours mathematical history on the study and formalism of the notion of nearness [37]. J. B. Listing and F. Hausdorff first considered the ideas of near and far within a mathematical context. They were followed by J. H. Poincaré's work on resemblance and similarity, which was inspired by kinesthesia experiments performed by E. Weber in 1834, and G. T. Fechner's sensation sensitivity experiments [39, 40, 41, 42]. Poincaré's reflections led him to introduce resemblance in representative spaces (called physical continua) [43, 44, 45], where his work marked a shift from stimuli and sensations to an abstraction in terms of mathematical sets together with an implicit idea of tolerance. Only a few years later, F. Riesz published a paper in 1908 on the nearness of sets [46, 47, 48], initiating the mathematical study of proximity spaces and the eventual discovery of descriptively near sets. During the 1960s, E.C. Zeeman introduced tolerance spaces in modelling visual perception [49], and A.B. Sossinsky observed in 1986 [50] that the main idea underlying tolerance space theory comes from Poincaré, especially [44]. The idea of near set theory – now known as descriptively near set theory – was inspired by a collaboration in 2002 by Z. Pawlak and J. F. Peters on a philosophic poem entitled “How Near” [51]. In 2006, a formal approach to the descriptive nearness of objects was considered by J. Peters, A. Skowron and J. Stepaniuk [52, 53] in the context of proximity spaces [47, 54, 48, 55]. This work led to the

²A nice history of near set theory is available in [37, 38].

introduction of descriptively near sets [56, 57], which was soon followed by tolerance near sets in 2009 [58, 59]. Tolerance near set theory builds on near set theory through the use of tolerance spaces and relations (based on work by Zeeman [60] and Sossinsky [50]³). Finally, the TNM was introduced in [36], which has been applied to the problems of image analysis and CBIR [62, 63, 64, 65, 66] and forms the main contribution of this article.

Finally, there is a large body of work on graph comparison methods other than those used in graph matching. This section concludes with recognition of these approaches. Tao et al. compare and contrast a graph measure defined with respect to topological indices with the GED [7]. Wallis et al. present an extension of the maximum common subgraph distance measure to propose a new graph distance metric. Similarly, [67] propose a new graph metric for measuring the similarity of graphs by identifying the relationship between the maximum common subgraph and the minimum common supergraph, and a new graph distance metric based on maximal common subgraph is presented by [68]. [2] present an approach for graph similarity scoring and matching based on structural similarity of local neighbourhoods. [69] demonstrate that graph similarity measures based on (sub)graph isomorphism and error-tolerant graph matching are a special case of a generic similarity measure proposed by [70]. A graph comparison methods for large graphs with hundreds of nodes is proposed in [71], and [72] present a graph similarity measure with low computational complexity based on the structural similarity of graphs. [73] propose a graph comparison measure based on the mutual information between graphs. Two further applications of graph comparison outside of graph matching are the work by [74] presenting an application of graph similarity measures for assessing attacks in dynamic networks, and an application of a pairwise graph comparison algorithm to the problem of automatic detection of conserved gene clusters reported by [75].

3. Graph Comparison Measures

The notion of quantifying the differences between two objects can be formalized through the definition of a similarity measure $d : X \times X \rightarrow \mathbb{R}$, where X is the representation space, containing a diversity of objects and their (multi-dimensional) descriptions, and \mathbb{R} is the set of real numbers. Further, d is considered a metric if conditions of non-negativity, uniqueness, symmetry and triangle inequality are satisfied. In this work, objects in X are attributed graphs and d provides a basis for decisions regarding the similarity of these graphs, which is known as graph comparison [1]. Here, the focus is on attributed graphs defined as a four tuple $G = (V, E, \alpha, \beta)$, where V is the set of vertices, E is a set of edges consisting of ordered pairs among elements $v \in V$, and α (resp. β) are functions that produce feature vectors containing information about node (edge) labels. With these definitions in mind, the remainder of this section describes the well-known GED metric and the recently reported tolerance nearness measure (TNM) for use in error-tolerant attributed graph matching.

³See, e.g., [61]

3.1. Graph Edit Distance

The graph edit distance (GED)⁴ is the most common and well know method for performing error-tolerant graph matching [10]. Briefly (and using the same notation as [76]), the GED is a measure of the amount of distortion required to transform one graph into another. Distortion actions, called edit operations, include insertion, deletion, and substitution of both nodes and edges. A series of distortions used to transition G^p into G^q is called an edit path, and is defined as $editPath(G^p, G^q) = (\epsilon_1, \dots, \epsilon_k)$, where ϵ_i represents edit operation i in this process of morphing one graph into another. The set of all edit paths for two graphs is labelled v , and for each $editPath \in v$ there is an associated bijection $f^{p,q} \in T$, where a bijection $f^{p,q} : V^p \rightarrow V^q$ assigns one vertex in G^p to only one vertex in G^q and T is the set of all possible bijections. Continuing, there is an associated cost function for each edit operation. Cost functions are combined to form an overall edit cost function, $EditCost(G^p, G^q, f^{p,q})$, used to find the minimum amount of operations required to transform G^p into G^q (defined in [77]) [76]. Finally, the GED is defined as $EditDistance(G^p, G^q) = \min_{f^{p,q} \in T} EditCost(G^p, G^q, f^{p,q})$.

The GED is generally determined by the A^* algorithm [78] when exactly evaluating the $EditDistance(G^p, G^q)$ [76]. However, the complexity of this approach is exponential in the number of nodes. As a result, approximate methods are necessary when working on anything but small examples. Approximate methods are appropriately named *sub-optimal* because they strike a balance between computational cost and non-exact GED values. While there are many sub-optimal algorithms (see, e.g., [76]), this article focuses on the bipartite algorithm for generating GED values since it is considered one of the best sub-optimal algorithms [76], and the results presented here were generated using the implementation reported in [10], which also uses this algorithm. There are three variants of the bipartite algorithm, namely the original approach reported by [27], fast bipartite matching [76], and, recently, the square fast bipartite matching [13]. The fast bipartite matching algorithm was used for this article. Lastly, given graphs G^p, G^q of order n, m , respectively, the time complexity for the bipartite algorithm is $O(n + m)^3$ and the time complexity for both the fast and square fast bipartite matching algorithms is $O(max(n, m)^3)$ [13].

3.2. Tolerance Nearness Measure

The focus of descriptive near set theory is to assess similarity in terms of the descriptions⁵ of objects within the sets. Until this article, the tolerance nearness measure (TNM) was used to quantify the similarity between two sets of objects, where these sets of objects were not graphs (see [65] for a detailed discussion). This process involves creating a single attributed graph, G^{XY} , based on the union of the two sets, $X, Y \subseteq O$, where the objects in O are used as graph vertices, i.e. $V^{XY} = X \cup Y$, and edges between vertices in V^{XY} are created based on the distance between feature vectors associated with each $v \in V^{XY}$. Specifically, a binary relation is defined between vertices based on their feature vectors, and edges between vertices are formed if the two objects satisfy the relation. Using G^{XY} , calculating the TNM requires finding all the maximal cliques, where, for each clique, the distribution of the objects between the original two sets is used to produce the final TNM value (see below).

⁴See [76] for an in-depth description of the GED.

⁵ [53] details the difference between traditional spatial nearness relations and relations defined in terms of descriptive nearness relations.

While useful, this process is not suited to assessing the similarity of graphs since new edges are produced during the calculation. For this reason, two approaches to using the TNM for quantifying the similarity of graphs are presented: one based solely on the vertices of the two graphs (using the original TNM approach), and the method presented below that incorporates the existing graph edge information in the calculation. Both of these approaches are graphically depicted in Figure 1.

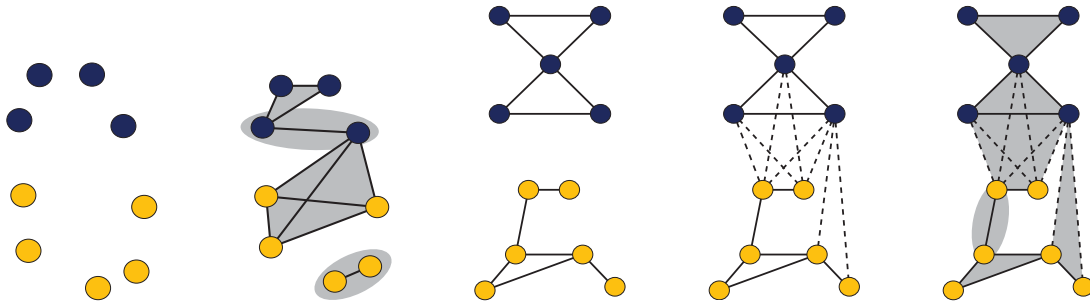


Figure 1. Visual example of two approaches for calculating the TNM. (a) and (b) represent the original approach, where (a) contains two sets of objects (represented by two colours) and (b) shows the edges created with a tolerance relation and the grey areas are maximal cliques. (c) - (e) depict the TNM in quantifying the similarity of graphs. (c) contains the original graphs (with edges), (d) shows new edges formed between the graphs using a tolerance relation, and the grey regions in (e) are maximal cliques.

Formally, the TNM is defined within the context of a perceptual system, which is denoted as $\langle O, \mathbb{F} \rangle$. A perceptual system consists of a non-empty set O of sample perceptual objects and a non-empty set \mathbb{F} of real-valued functions $\phi \in \mathbb{F}$, where $\phi : O \rightarrow \mathbb{R}$ [79, 36]. Recall, for this application, objects in O are graph vertices, and, for an attributed graph, each vertex has an associated feature vector generated by α . Within a perceptual system, α is defined as a feature vector given by $\alpha(x) = (\phi_1(x), \phi_2(x), \dots, \phi_i(x), \dots, \phi_l(x))$, where l is the length of the vector α , and each $\phi_i(x)$ in $\alpha(x)$ is a probe function value [80] forming part of the description of $x \in O$. In the descriptive near set literature, the feature vector α is called an object description.

The first step in finding the TNM between two sets of objects is the creation of G^{XY} using a binary tolerance relation [61, 36]. Briefly, a tolerance relation is one that is reflexive and symmetric, but not transitive. The specific relation used in this work is defined as follows. Given $\varepsilon \in \mathbb{R}_0^+$ and \mathbb{F} , the perceptual tolerance relation \cong_ε is defined as

$$\cong_\varepsilon = \{(x, y) \in O \times O : \|\alpha(x) - \alpha(y)\|_2 \leq \varepsilon\}, \tag{1}$$

where $\|\cdot\|_2$ is the L^2 norm [36]. Using \cong_ε , E^{XY} is formed by adding an edge between $x, y \in V^{XY}$ if, and only if, (iff) two objects satisfy this relation, *i.e.* the distance between the two feature vectors is less than ε . Fig. 1(a) & 1(b) give an example of forming edges in this manner. Now that $G^{XY} =$

$(V^{XY}, E^{XY}, \alpha, \beta)$ has been defined⁶ the TNM, between sets X and Y , is given as

$$TNM(G^{XY}, V^X, V^Y) = 1 - \left(\sum_{C \in H} |C| \right)^{-1} \cdot \sum_{C \in H} |C| \frac{\min(|C \cap V^X|, |C \cap V^Y|)}{\max(|C \cap V^X|, |C \cap V^Y|)}, \quad (2)$$

where H is the set of all maximum cliques obtained from G^{XY} , *i.e.* each $C \in H$ is a set of vertices representing a maximal clique in G^{XY} . Note, implicit in this definition is the requirement that H must be extracted from G^{XY} prior to performing the calculation given in Eq. 2.

As highlighted in the introduction, the focus of this article is the observation that Eq. 2 can also be used to assess the similarity between two graphs G^p and G^q . In this case, G^{pq} is formed in the following manner. The set of vertices is simply $V^{pq} = V^p \cup V^q$. E^{pq} , on the other hand, is obtained by the following two steps. First, set $E^{pq} = E^p \cup E^q$. Then, each $v_p \in V^p$ is compared to each $v_q \in V^q$, and an edge is only added to E^{pq} iff $v_p \cong_{\epsilon} v_q$. Creating E^{pq} in this manner is illustrated in Fig. 1(c) - 1(e), and Algorithm 1 contains the pseudocode for calculating the TNM using this approach. Note, edges are only added between the graphs, not between vertices within the same graph. Also, $MCE_{BK}(V^{pq}, E^{pq})$ performs maximal clique enumeration using the Bron-Kerbosh algorithm [81], and it returns a collection (family) of sets, each consisting of the vertices contained in a maximal clique. In Line 8, the cliques are chosen from H in the order in which they are generated by the MCE algorithm $MCE_{BK}(V^{pq}, E^{pq})$.

Notice that the complexity for calculating TNM depends on the approach used to find all the maximal cliques in a graph. Specifically, the Bron-Kerbosh algorithm [81] was used in this work to produce H . While the results reported here use the original Bron-Kerbosh algorithm, the addition of adding pruning methods gives this algorithm a worst-case running time complexity of $O(3^{n/3})$ for a graph of n vertices. This is optimal since there exists up to $3^{n/3}$ maximal cliques in a graph with n vertices [82]. However, the TNM algorithm also consists of either: finding edges between all nodes when comparing two sets of objects (of sizes n and m), or only finding edges between the two sets of vertices when comparing graphs (also of sizes n and m). Thus, the final worst case runtime complexities are $O((n+m)^2 + 3^{(n+m)/3}) = O((n+m)^2)$ and $O(\max(n, m)^2 + 3^{(n+m)/3}) = O(\max(n, m)^2)$, respectively. Observe that both of these complexities are lower than those reported for the GED above.

4. Results and Discussion

This section compares the GED and TNM through their use in performing content-based image retrieval (CBIR) on two different datasets. CBIR is the task of retrieving all the images from a digital image dataset that match a given query image. The goal is to retrieve all the perceptually similar images, and in most cases rank the relevance of each image to the query image. Besides subjective evaluation of the results, CBIR methods are typically compared using datasets partitioned into cat-

⁶ β is not used in this work, *i.e.* there is no extra information associated with edges.

Algorithm 1: TNM(edges) Algorithm

Input : G^p, G^q, ε
Output: $TNM(G^{pq}, V^p, V^q)$
 $V^{pq} \leftarrow V^p \cup V^q; E^{pq} \leftarrow E^p \cup E^q;$
for $v_p \in V^p$ **do**
 | **for** $v_q \in V^q$ **do**
 | | **if** $v_p \cong_{\varepsilon} v_q$ **then**
 | | | $E^{pq} \leftarrow E^{pq} \cup (v_p, v_q);$
 | | **end**
 | **end**
end
 $H = \text{MCE}_{\text{BK}}(V^{pq}, E^{pq});$
 $w \leftarrow 0; TNM \leftarrow 0;$
for $C \in H$ **do**
 | $w \leftarrow w + |C|;$
 | $TNM = TNM + |C| \frac{\min(|C \cap V^p|, |C \cap V^q|)}{\max(|C \cap V^p|, |C \cap V^q|)};$
end
 $TNM \leftarrow 1 - TNM/w;$

egories of images. Taking this approach allows the use of metrics such as precision and recall to evaluate, compare, and contrast the reported methods.

In comparing the two measures in this article, test datasets used in previously published GED and TNM papers were employed. Consequently, the datasets used in this work are the Tarragona Rotation Zoom [3, 4], SIMPLicity [5] and the Palmprint datasets [6]. The Tarragona Rotation Zoom dataset was selected as this article uses the same GED implementation from [10] – as well as the learnt edit-costs identified in the paper. SIMPLicity was similarly chosen for its use in earlier TNM papers [36, 65]. The Palmprint dataset was picked to demonstrate the increase in computational complexity as the size of the graph grows.

The Tarragona dataset consists of four categories: boat, east park, east south, and residence. Each category consists of 10 images, where each image produces a single graph. The features associated with each vertex in a graph are extracted using the SpeedUp Robust Features (SURF) algorithm [83] and the edges between the vertices are created using Delaunay triangulation [84]. Each graph consists of 50 vertices and 64 features. There are no attributes associated with graph edges.

The SIMPLicity dataset contains 10 categories, where each category consists of 100 images. Again, a single graph is extracted from each image, consisting of 228 vertices and 18 features per vertex. Features are extracted using texture based techniques, moments in CIELUV colour space, edge based features and Zernike moments [36]. Edges between vertices are created using Eq. 1, where an edge is formed if two vertices satisfy the perceptual tolerance relation. The graphs used to generate the results were created using $\varepsilon = 0.2^7$ as this value produced the best results in [65]. The

⁷See [36] for a discussion on how the TNM varies with ε .

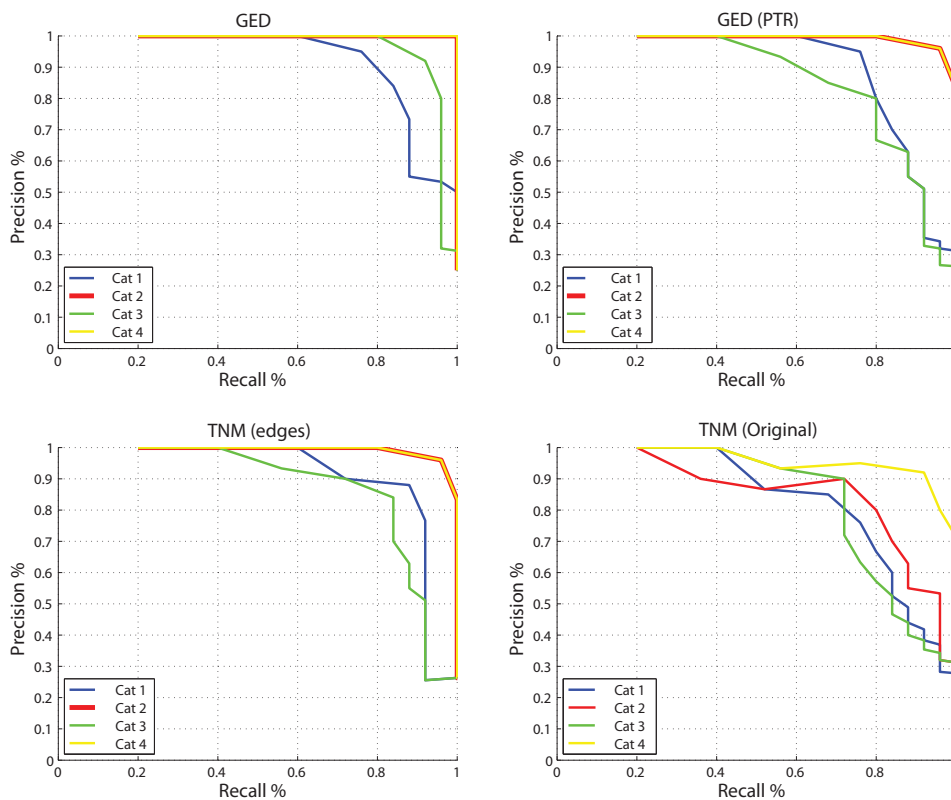


Figure 2. Average precision vs. recall plots for the GED and TNM on the Tarragona dataset. (a) GED, (b) GED (PTR), (c) TNM (edges), and (d) TNM (Original).

value of ε is typically selected based on trial and error and falls into the range $[0, \sqrt{l}]$, where l is the feature vector length, $\varepsilon = 0$ produces no edges and $\varepsilon = \sqrt{l}$ produces a fully connected graph.

A subset of the Palmpoint dataset is extracted because of very high computational time which contains 4 categories and each category consists of 8 images (graphs). The dataset contains variable number of nodes. Maximum nodes in a graph are 1385 and minimum is 574, where the average number of nodes is ~ 841 . There are total 5 features associated with each nodes *i.e.* position, type, quality and type of minutiae. The features associated with each node in a graph are extracted using the algorithm described in [85] and the original edges between the nodes were created using delaunay triangulation with no attributes [6]. The edges used in this experiment are created using the perceptual tolerance relation with $\varepsilon = 0.4$.

The results are compared in this article using precision and recall [86] and are given in Tables 1 & 2 & 3 and Figures 2 & 3 & 4 Here it is important to note a few points. First, the TNM results presented here are broken into two categories, namely the original approach (depicted in Figs. 1(a) & 1(b)) and the method that incorporates the existing graph edge information in the calculation (depicted in Figs. 1(c)-1(e)). These two approaches are labelled *original* and *edges*. Next, with respect to the SIMPLICity & Palmpoint dataset, the TNM *original* and *edges* results are the same

since the SIMPLicity & Palmprint graphs are created with the perceptual tolerance relation. Thus, only one set of figures is presented. Lastly, the Tarragona dataset consists of graphs, however, for comparison with the TNM, new edges are also created between vertices using the perceptual tolerance relation. These results are labelled *GED (perceptual tolerance relation (PTR))* and were generated using $\varepsilon = 1.5$. This value of epsilon was selected as it produces graphs which generate the same final learnt costs when using the approach reported in [10].

Table 1. Comparison of GED and TNM on Tarragona dataset using Precision(P) and Recall(R)

Category	GED		GED (PTR)		TNM (edges)		TNM (Original)	
	P	R	P	R	P	R	P	R
Boat	0.9500	0.7600	0.9500	0.7600	0.9000	0.7200	0.8500	0.6800
East Park	1.000	0.8000	1.000	0.8000	1.000	0.8000	0.9000	0.7200
East South	1.000	0.8000	0.8500	0.6800	0.9000	0.7200	0.9000	0.7200
Residence	1.000	0.8000	1.000	0.8000	1.000	0.8000	0.9500	0.7600
Average	0.9875	0.7900	0.9500	0.7600	0.9500	0.7600	0.9000	0.7200
Timing (ms)	43.33		45.71		40.29		16.29	

Table 2. Comparison of GED and TNM on SIMPLicity dataset using Precision(P) and Recall(R)

Category	GED		TNM	
	P	R	P	R
1. Africans	0.6635	0.1327	0.7230	0.1446
2. Beaches	0.5245	0.1049	0.4755	0.0951
3. Buildings	0.4385	0.0877	0.4745	0.0949
4. Buses	0.6985	0.1397	0.5145	0.1029
5. Di-nosaurs	1.000	0.2000	1.000	0.2000
6. Ele-phants	0.7680	0.1536	0.7210	0.1442
7. Flowers	0.9470	0.1894	0.8615	0.1723
8. Horses	0.8375	0.1675	0.7360	0.1472
9. Moun-tains	0.6425	0.1285	0.6325	0.1265
10. Food	0.7130	0.1426	0.7425	0.1485
Average	0.7233	0.14466	0.6881	0.13762
Timing (ms)	11.36		0.61	

The results were generated as follows. First, a single image was selected as the query image, which

Table 3. Comparison of GED and TNM on Palmprint dataset using Precision(P) and Recall(R)

Category	GED		TNM	
	P	R	P	R
1. Cat 1	0.6875	0.3438	0.6563	0.3281
2. Cat 2	0.4063	0.2031	0.5625	0.2813
3. Cat 3	0.5938	0.2969	0.6875	0.3438
4. Cat 4	1	0.5000	1	0.5000
Average	0.6719	0.3359	0.7265	0.3633
Timing (s)	1127.348		283.144	

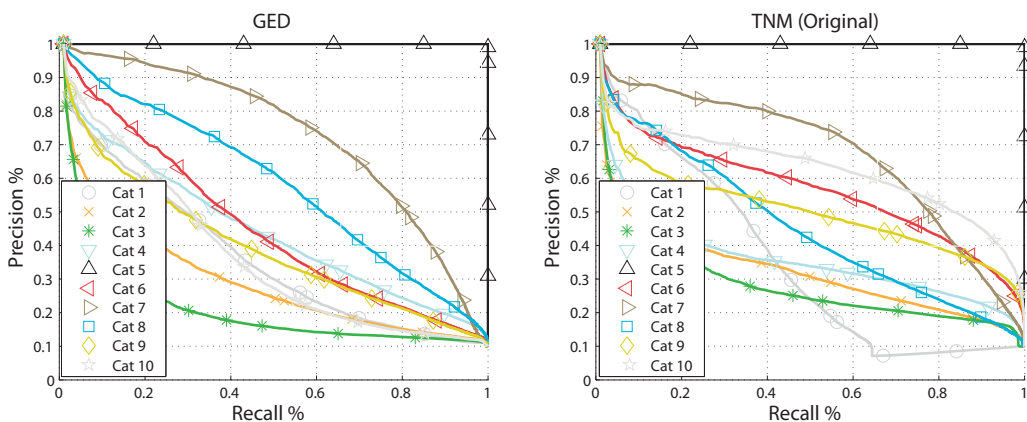


Figure 3. Average precision vs. recall plots for the GED and TNM on the SIMPLicity dataset. (a) GED, and (b) TNM.

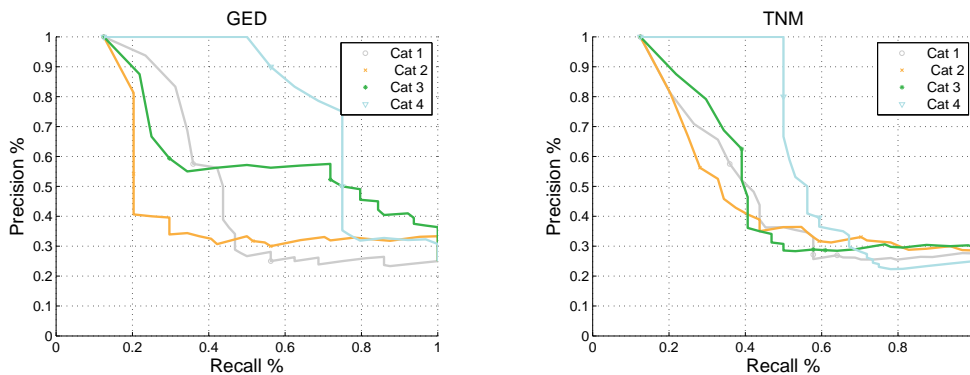


Figure 4. Average precision vs. recall plots for the GED and TNM on the Palmprint dataset. (a) GED, and (b) TNM.

was compared to each other image in the dataset using either the GED or TNM approaches mentioned above. The measure values were then sorted and the image category information was used to calculate precision and recall. This process was then repeated for each image in the dataset. Finally, precision and recall values were averaged across each category and the results are given in Figures 2 & 3 & 4. Similarly, the results reported in Tables 1 & 2 & 3 represent the values obtained for the top 4, top 20 and top 4 images, respectively, in each category. Note, this process consisted of 210 comparisons for the Tarragon dataset, 500,500 comparisons for SIMPLIcity and 528 comparisons for the Palmprint.

Given this information, some interesting observations about the results can be made. Beginning with Tables 1 & 2 notice that the TNM results are all quite similar to the GED. With respect to the original TNM (using only vertex information) and the Tarragona dataset, TNM precision values are lower than both GED and GED (PTR)⁸. Recall values are closer than precision, but still all lower than GED. The TNM using the graph edge information performs better, where precision and recall values tie GED in East Park and Residence categories. The better performance makes sense since more of the original, domain-specific graph information is used. With respect to the SIMPLIcity dataset, TNM outperforms GED in the Africans, Buildings, and Food categories. As before, GED performs better in the majority of cases, but the results are close. With respect to Figures 2 & 3, it is harder to make a direct comparison. In general, Figure 2 shows that TNM produces comparable results to the GED, where GED performs slightly better. On the other hand, from Figure 3, TNM performs better in the Beaches, Buildings, Elephants, and Mountains categories, while GED performs better in Buses, Horses, and Food categories. Both measures perform comparably in categories Africans, Dinosaurs, and Flowers. Table 3 shows the results for the Palmprint dataset for both TNM and GED; where GED performs slightly better than TNM in first category. On the contrary, the TNM performs better than GED in second and third category and the GED & the TNM tie results for fourth category. The overall average precision and recall represents that the TNM performs better than the GED with much less computational time for single comparison. Figure 4 shows the results for both TNM and GED are very close, however, the TNM performs better in most of the categories.

The average time per graph comparison for each approach are given in the bottom of Tables 1 & 2 & 3. Notice, as expected, the runtime for the TNM approach is less than the GED approach in all cases. Further, the TNM(Original) approach is much faster than the TNM(edges) approach since the graphs produced by the perceptual tolerance relation with $\epsilon = 1.5$ contain less edges than the original Tarragona graphs. Likewise, the timing results presented in Table 3 shows that computational time increases with the increase in size of graph which in this case is much less for the TNM than the GED. Finally, these timing results were generated using Matlab version R2012a on a system running Windows 10 with an Intel i5-4200U CPU @ 1.6GHZ and 4GB or memory.

5. Conclusion

This article presented a new application of the TNM for use as a graph comparison measure, and demonstrated it is a competitive alternative to the popular and well-known GED. Graph-based results were presented with respect to the original TNM definition (using only vertices in the calculation) as

⁸Recall, GED (PTR) uses the same edges as used in the TNM calculation (rather than the edges that are present in the dataset).

well as a new approach that also uses graph edge information. The results demonstrate that both measures produced comparable precision and recall values. Further, the TNM approach is advantageous in that it is not dependent on costs, *i.e.* cost functions must be defined for each edit operation in the GED, whereas the TNM approach is only dependent on ε from Eq. 1. Additionally, the runtime complexity of the TNM was shown to be less than that of GED, which is important as graph sizes grow ever larger. Future work will consist of exploration into learning optimal ε values for a given problem domain, and applications of these methods to other datasets since results suggest TNM may perform as good (or better) than GED in many real world applications.

References

- [1] Riesen K. Structural Pattern Recognition with Graph Edit Distance: Approximation Algorithms and Applications. Springer, 2016.
- [2] Zager LA, Verghese GC. Graph similarity scoring and matching. *Applied Mathematics Letters*, 2008. **21**(1):86 – 94.
- [3] Cortés X, Serratos F. Learning Graph-matching Edit-costs Based on the Optimality of the Oracle’s Node Correspondences. *Pattern Recogn. Lett.*, 2015. **56**(C):22–29.
- [4] Corts X, Serratos F. An interactive method for the image alignment problem based on partially supervised correspondence. *Expert Systems with Applications*, 2015. **42**(1):179 – 192.
- [5] Wang JZ, Li J, Wiederhold G. SIMPLiCity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001. **23**(9):947–963.
- [6] Moreno-García CF, Serratos F. Correspondence consensus of two sets of correspondences through optimisation functions. *Pattern Analysis and Applications*, 2017. **20**(1):201–213. doi:10.1007/s10044-015-0486-y. URL <https://doi.org/10.1007/s10044-015-0486-y>.
- [7] Li T, Dong H, Shi Y, Dehmer M. A comparative analysis of new graph distance measures and graph edit distance. *Information Sciences*, 2017. **403404**:15 – 21.
- [8] Conte D, Foggia P, Sansone C, Vento M. THIRTY YEARS OF GRAPH MATCHING IN PATTERN RECOGNITION. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004. **18**(03):265–298.
- [9] Serratos F, Cortés X. Interactive graph-matching using active query strategies. *Pattern Recognition*, 2015. **48**(4):1364–1373.
- [10] Cortés X, Serratos F. Learning graph-matching edit-costs based on the optimality of the oracle’s node correspondences. *Pattern Recognition Letters*, 2015. **56**:22–29.
- [11] Conte D, Ramel JY, Sidre N, Luqman MM, Gaüzère B, Gibert J, Brun L, Vento M. A Comparison of Explicit and Implicit Graph Embedding Methods for Pattern Recognition. In: Kropatsch WG, Artner NM, Haxhimusa Y, Jiang X (eds.), Graph-Based Representations in Pattern Recognition: 9th IAPR-TC-15 International Workshop, GbRPR 2013, Vienna, Austria, May 15-17, 2013. Proceedings, pp. 81–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [12] Fischer A, Suen CY, Frinken V, Riesen K, Bunke H. A Fast Matching Algorithm for Graph-Based Handwriting Recognition, pp. 194–203. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

- [13] Serratos F. Speeding up Fast Bipartite Graph Matching Through a New Cost Matrix. *International Journal of Pattern Recognition and Artificial Intelligence*, 2015. **29**(2):1550010–1 – 1550010–17.
- [14] Riesen K, Bunke H. Reducing the dimensionality of dissimilarity space embedding graph kernels. *Engineering Applications of Artificial Intelligence*, 2009. **22**(1):48–56.
- [15] Gaüzère B, Hasegawa M, Brun L, Tabbone S. Implicit and Explicit Graph Embedding: Comparison of Both Approaches on Chemoinformatics Applications. In: Gimelfarb G, Hancock E, Imiya A, Kuijper A, Kudo M, Omachi S, Windeatt T, Yamada K (eds.), *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR&SPR 2012, Hiroshima, Japan, November 7-9, 2012*. Proceedings, pp. 510–518. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [16] Bunke H, Riesen K. Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, 2011. **44**(5):1057–1067.
- [17] Zaslavskiy M, Bach F, Vert JP. Global alignment of proteinprotein interaction networks by graph matching methods. *Bioinformatics*, 2009. **25**(12):i259–1267.
- [18] Cosmo L, Rodola E, Albarelli A, Mmoli F, Cremers D. Consistent partial matching of shape collections via sparse modeling. In: *Computer Graphics Forum*. Wiley Online Library .
- [19] Bergamasco F, Albarelli A, Torsello A. A graph-based technique for semi-supervised segmentation of 3D surfaces. *Pattern Recognition Letters*, 2012. **33**(15):2057–2064.
- [20] Sanfeliu A, Fu KS. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983. **SMC-13**(3):353–362.
- [21] Bunke H, Allermann G. Inexact graph matching for structural pattern recognition. *Pattern Recogn. Lett.*, 1983. **1**(4):245–253.
- [22] Solé A, Serratos F, Sanfeliu A. On the graph edit distance cost: properties and applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 2012. **26**:24.
- [23] Gao X, Xiao B, Tao D, Li X. A survey of graph edit distance. *Pattern Analysis and Applications*, 2010. **13**(1):113–129.
- [24] Morgan HL. Spelling correction in systems programs. *Commun. ACM*, 1970. **13**(2):90–94.
- [25] Wagner RA, Fischer MJ. The String-to-String Correction Problem. *J. ACM*, 1974. **21**(1):168–173.
- [26] Tai KC. The Tree-to-Tree Correction Problem. *J. ACM*, 1979. **26**(3):422–433.
- [27] Riesen K, Bunke H. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 2009. **27**(7):950–959.
- [28] Neuhaus M, Bunke H. An Error-Tolerant Approximate Matching Algorithm for Attributed Planar Graphs and Its Application to Fingerprint Classification. In: Fred A, Caelli TM, Duin RPW, Campilho AC, de Ridder D (eds.), *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004*, pp. 180–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [29] Robles-Kelly A, Hancock ER. Graph edit distance from spectral seriation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. **27**(3):365–378.
- [30] Justice D, Hero A. A binary linear programming formulation of the graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. **28**(8):1200–1214.

- [31] Marn RM, Aguirre NF, Daza EE. Graph Theoretical Similarity Approach To Compare Molecular Electrostatic Potentials. *Journal of Chemical Information and Modeling*, 2008. **48**(1):109–118.
- [32] Ambauen R, Fischer S, Bunke H. Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification, pp. 95–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [33] Zheng W, Zou L, Lian X, Wang D, Zhao D. Graph similarity search with edit distance constraint in large graph databases. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, CIKM '13. ACM, New York, NY, USA. ISBN 978-1-4503-2263-8, 2013 pp. 1595–1600.
- [34] Ibragimov R, Malek M, Guo J, Baumbach J. GEDEVO: An Evolutionary Graph Edit Distance Algorithm for Biological Network Alignment. In: Beißbarth T, Kollmar M, Leha A, Morgenstern B, Schultz AK, Waack S, Wingender E (eds.), German Conference on Bioinformatics 2013, volume 34 of *OpenAccess Series in Informatics (OASISs)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013 pp. 68–79.
- [35] Fischer A, Suen CY, Frinken V, Riesen K, Bunke H. Approximation of graph edit distance based on Hausdorff matching. *Pattern Recognition*, 2015. **48**(2):331 – 343.
- [36] Henry CJ. Perceptually Indiscernibility, Rough Sets, Descriptively Near Sets, and Image Analysis. *Transactions on Rough Sets*, 2012. **LNCS 7255**:41–121.
- [37] Peters JF, Naimpally SA. Applications of near sets. *Notices of the American Mathematical Society*, 2012. **59**(4):536–542.
- [38] Peters JF. Near sets. *Wikipedia, The Free Encyclopaedia*, 2015. Edited by C. J. Henry.
- [39] Poincaré H. Science and Hypothesis. The Mead Project, Brock University, 1905. L. G. Ward's translation.
- [40] Benjamin LT Jr. A Brief History of Modern Psychology. Blackwell Publishing, Malden, MA, 2007.
- [41] Hergenhahn BR. An Introduction to the History of Psychology. Wadsworth Publishing, Belmont, CA, 2009.
- [42] Fechner GT. Elements of Psychophysics, vol. I. Hold, Rinehart & Winston, London, UK, 1966. H. E. Adler's trans. of Elemente der Psychophysik, 1860.
- [43] Poincaré H. Sur la nature of raisonnement mathématique. *Revue de métaphysique et de morale*, 1894. **2**:371–384.
- [44] Poincaré H. L'espace et la géométrie. *Revue de métaphysique et de morale*, 1895. **3**:631–646.
- [45] Poincaré H. Sur certaines surfaces algébriques; troisième complément a l'analysis situs. *Bulletin de la Société de France*, 1902. **30**:49–70.
- [46] Riesz F. Stetigkeitsbegriff und abstrakte mengenlehre. *Atti del IV Congresso Internazionale dei Matematici*, 1908. **II**:18–24.
- [47] Naimpally SA, Warrack BD. Proximity Spaces. In: Cambridge Tract in Mathematics No. 59. Cambridge University Press, Cambridge, UK, 1970.
- [48] Naimpally SA. Near and Far. A centennial Tribute to Frigyes Riesz. *Siberian Electronic Mathematical Reports*, 2009. **6**:A.1–A.10.
- [49] Zeeman EC. The topology of the brain and the visual perception. In: Fort KM (ed.), Topology of 3-manifolds and selected topics, pp. 240–256. University of Georgia Institute Conference Proceedings, Prentice Hall, Georgia, 1962.

- [50] Sossinsky AB. Tolerance space theory and some applications. *Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications*, 1986. **5**(2):137–167.
- [51] Pawlak Z, Peters JF. Jak blisko (how near). *Systemy Wspomagania Decyzji*, 2002. **1**:57–109.
- [52] Peters JF. Near Sets: An Introduction. *Mathematics in Computer Science*, 2013. **7**(1):3–9.
- [53] Naimpally SA, Peters JF. Topology with Applications. Topological Spaces via Near and Far. World Scientific, Singapore, 2013.
- [54] Mozzochi CJ, Naimpally SA. Uniformity and Proximity. In: Allahabad Mathematical Society Lecture Note Series, volume 2. The Allahabad Math. Soc., 2009 p. 153 pp.
- [55] Hocking JG, Naimpally SA. Nearness-A Better Approach to Continuity and Limits. In: Allahabad Mathematical Society Lecture Note Series, volume 3. The Allahabad Math. Soc., 2009 p. 153 pp.
- [56] Peters JF. Near Sets. General Theory About Nearness of Objects. *Applied Mathematical Sciences*, 2007. **1**(53):2609–2629.
- [57] Peters JF. Near sets. Special theory about nearness of objects. *Fundamenta Informaticae*, 2007. **75**(1-4):407–433.
- [58] Peters JF. Tolerance near sets and image correspondence. *International Journal of Bio-Inspired Computation*, 2009. **1**(4):239–245.
- [59] Peters JF. Corrigenda and addenda: Tolerance near sets and image correspondence. *International Journal of Bio-Inspired Computation*, 2010. **2**(5):310–318.
- [60] Zeeman EC. The topology of the brain and the visual perception. In: Fort KM (ed.), Topology of 3-manifolds and selected topics, pp. 240–256. Prentice Hall, New Jersey, 1965.
- [61] Peters JF, Wasilewski P. Tolerance spaces: Origins, theoretical aspects and applications. *Information Sciences*, 2012. **195**(0):211–225.
- [62] Henry C, Peters JF. Perceptual Image Analysis. *International Journal of Bio-Inspired Computation*, 2010. **2**(3/4):271–281.
- [63] Henry C, Peters JF. Perception-Based Image Classification. *International Journal of Intelligent Computing and Cybernetics*, 2010. **3**(3):410–430. Emerald Literati Network 2011 Award for Excellence.
- [64] Henry C, Peters JF. Arthritic hand-finger movement similarity measurements: Tolerance near set approach. *Computational and Mathematical Methods in Medicine*, 2011. **2011**. Article ID 569898, 14 pp.
- [65] Henry CJ, Ramanna S. Maximal Clique Enumeration in Finding Near Neighbourhoods, pp. 103–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [66] Henry CJ, Ramanna S. Signature-based Perceptual Nearness. Application of Near Sets to Image Retrieval. *Mathematics in Computer Science*, 2013. **7**(1):71–85.
- [67] Fernandez ML, Valiente G. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 2001. **22**(67):753 – 758.
- [68] Bunke H, Shearer K. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 1998. **19**(34):255 – 259.

- [69] Sorlin S, Solnon C. Reactive Tabu Search for Measuring Graph Similarity, pp. 172–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [70] Champin PA, Solnon C. Measuring the similarity of labeled graphs. In: International Conference on Case-Based Reasoning. Springer, 2003 pp. 80–95.
- [71] Shervashidze N, Vishwanathan S, Petri T, Mehlhorn K, Borgwardt K. Efficient graphlet kernels for large graph comparison. In: Artificial Intelligence and Statistics. 2009 pp. 488–495.
- [72] Dehmer M, Emmert-Streib F, Kilian J. A similarity measure for graphs with low computational complexity. *Applied Mathematics and Computation*, 2006. **182**(1):447 – 459.
- [73] The mutual information between graphs. *Pattern Recognition Letters*, 2017. **87**:12 – 19. Advances in Graph-based Pattern Recognition.
- [74] Bopche GS, Mehtre BM. Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks. *Computers & Security*, 2017. **64**:16 – 43.
- [75] Fujibuchi W, Ogata H, Matsuda H, Kanehisa M. Automatic detection of conserved gene clusters in multiple genomes by graph comparison and P-quasi grouping. *Nucleic Acids Research*, 2000. **28**(20):4029.
- [76] Serratoso F. Fast computation of Bipartite graph matching. *Pattern Recognition Letters*, 2014. **45**:244–250.
- [77] Bunke H. Error correcting graph matching: on the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999. **21**(9):917–922.
- [78] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968. **4**(2):100–107.
- [79] Peters JF, Wasilewski P. Foundations of Near Sets. *Info. Sci.*, 2009. **179**(18):3091–3109.
- [80] Pavel M. Fundamentals of Pattern Recognition. Marcel Dekker, Inc., NY, 1993.
- [81] Bron C, Kerbosch J. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 1973. **16**(9):575–577.
- [82] Tomita E, Tanaka A, Takahashi H. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 2006. **363**(1):28–42.
- [83] Zhou M, Asari VK. A Fast Video Stabilization System Based on Speeded-up Robust Features, pp. 428–435. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [84] Overmars M, De Berg M, van Kreveld M, Schwarzkopf O. Computational Geometry: Algorithms and Applications. Springer-Verlag, Berlin, Heidelberg, 2008.
- [85] Dai J, Zhou J. Multifeature-Based High-Resolution Palmprint Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. **33**(5):945–957. doi:10.1109/TPAMI.2010.164.
- [86] Yates-Baeza R, Ribeiro-Neto B. Modern Information Retrieval. ACM Pres/Pearson Addison Wesley, New York, 1999.
- [87] Alusaifeer T, Ramanna S, Henry CJ, Peters J. GPU Implementation of MCE approach to Finding Near Neighbourhoods. *Lecture Notes in Artificial Intelligence*, 2013. **LNAI 8171**:251–262.
- [88] Basak SC, Magnuson VR, Niemi GJ, Regal RR. Determining structural similarity of chemicals using graph-theoretic indices. *Discrete Applied Mathematics*, 1988. **19**(1):17–44.

- [89] Bunke H. Graph matching: Theoretical foundations, algorithms, and applications. In: Proceedings of Vision Interface 2000. 2000 pp. 82–88.
- [90] Cantoni V, Cinque L, Guerra C, Levialdi S, Lombardi L. 2-D OBJECT RECOGNITION BY MULTI-SCALE TREE MATCHING. *Pattern Recognition*, 1998. **31**(10):1443–1454.
- [91] Cook DJ, Holder LB. Substructure discovery using minimum description length and background knowledge. *J. Artif. Int. Res.*, 1994. **1**(1):231–255.
- [92] Ehrig H, Habel A, Kreowski HJ. Introduction to graph grammars with applications to semantic networks. *Computers & Mathematics with Applications*, 1992. **23**(6):557–572.
- [93] Fernandez ML, Valiente G. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 2001. **22**(67):753–758.
- [94] Henry CJ. A Parallel GPU Solution to the Maximal Clique Enumeration Problem for CBIR. In: GPU Technology Conference (GTC 2014). 2014 .
- [95] Irniger C, Bunke H. Theoretical Analysis and Experimental Comparison of Graph Matching Algorithms for Database Filtering. In: Hancock E, Vento M (eds.), Graph Based Representations in Pattern Recognition: 4th IAPR International Workshop, GbRPR 2003 York, UK, June 30 July 2, 2003 Proceedings, pp. 118–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [96] Jonker R, Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 1987. **38**(4):325–340.
- [97] Kuhn HW. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955. **2**(1-2):83–97.
- [98] Levi G. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *CALCOLO*, 1973. **9**(4):341–352.
- [99] Li J, Wang JZ. Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003. **25**(9):1075–1088.
- [100] Maher PE. A similarity measure for conceptual graphs. *International Journal of Intelligent Systems*, 1993. **8**(8):819–837.
- [101] McGregor JJ. Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, 1982. **12**(1):23–34.
- [102] Messmer BT, Bunke H. Automatic learning and recognition of graphical symbols in engineering drawings. In: Kasturi R, Tombre K (eds.), Graphics Recognition Methods and Applications: First International Workshop University Park, PA, USA, August 1011, 1995 Selected Papers, pp. 123–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.
- [103] Munkres J. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 1957. **5**(1):32–38.
- [104] Raveaux R, Burie JC, Ogier JM. A graph matching method and a graph matching distance based on subgraph assignments. *Pattern Recognition Letters*, 2010. **31**(5):394–406.
- [105] Robles-Kelly A, Hancock ER. Graph Matching Using Spectral Seriation and String Edit Distance. In: Hancock E, Vento M (eds.), Graph Based Representations in Pattern Recognition: 4th IAPR International Workshop, GbRPR 2003 York, UK, June 30 July 2, 2003, pp. 154–165. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

- [106] Rocha J, Pavlidis T. A shape analysis model with applications to a character recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. **16**(4):393–404.
- [107] Sanfeliu A, Alquézar R, Andrade J, Climent J, Serratosa F, Vergés J. Graph-based representations and techniques for image processing and image analysis. *Pattern Recognition*, 2002. **35**(3):639–650.
- [108] Schenker A, Bunke H, Last M, Kandel A. Polynomial Time Complexity Graph Distance Computation for Web Content Mining. In: Basu M, Ho TK (eds.), *Data Complexity in Pattern Recognition*, pp. 197–215. Springer London, London, 2006.
- [109] Schenker A, Last M, Bunke H, Kandel A. Classification of Web documents using a graph model. In: *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. 2003 pp. 240–244 vol.1.
- [110] Schenker A, Last M, Bunke H, Kandel A. CLASSIFICATION OF WEB DOCUMENTS USING GRAPH MATCHING. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004. **18**(03):475–496.
- [111] Serratosa F, Cortés X. Graph Edit Distance: Moving from global to local structure to solve the graph-matching problem. *Pattern Recognition Letters*, 2015. **65**:204–210.
- [112] Shearer K, Bunke H, Venkatesh S. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 2001. **34**(5):1075–1091.
- [113] Si Wei L, Ren Y, Suen CY. Hierarchical attributed graph representation and recognition of handwritten chinese characters. *Pattern Recognition*, 1991. **24**(7):617–632.
- [114] Wallis WD, Shoubridge P, Kraetz M, Ray D. Graph distances using graph union. *Pattern Recognition Letters*, 2001. **22**(67):701–704.
- [115] Wilson R, Beineke L. *Applications of graph theory*. Academic Press, 1979.
- [116] Wong AKC, You M. Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985. **PAMI-7**(5):599–609.
- [117] Henry CJ. *Near sets: theory and applications*. Ph.D. thesis, University of Manitoba, 2010.