Bayesian Graph Edit Distance

Richard Myers, Richard C. Wilson, and
Edwin R. Hancock

Abstract—This paper describes a novel framework for comparing and matching corrupted relational graphs. The paper develops the idea of edit-distance originally introduced for graph-matching by Sanfeliu and Fu [1]. We show how the Levenshtein distance can be used to model the probability distribution for structural errors in the graph-matching problem. This probability distribution is used to locate matches using MAP label updates. We compare the resulting graph-matching algorithm with that recently reported by Wilson and Hancock. The use of edit-distance offers an elegant alternative to the exhaustive compilation of label dictionaries. Moreover, the method is polynomial rather than exponential in its worst-case complexity. We support our approach with an experimental study on synthetic data and illustrate its effectiveness on an uncalibrated stereo correspondence problem. This demonstrates experimentally that the gain in efficiency is not at the expense of quality of match.

Index Terms—Graph matching, edit-distance, Bayesian, MAP estimation, stereo images.

1 Introduction

Relational matching is an abstract problem which models many processes in machine vision, ranging from midlevel tasks such as stereopsis [2] and multiple sensor fusion [3], [4] to higher-level ones such as object recognition [5] and scene understanding [6]. The two key issues in graph-matching are how to measure similarity when structural corruption is present and how to search efficiently for the best match. The first of these problems was extensively addressed in the structural pattern recognition literature of the 1980s. Concrete examples include Shapiro and Haralick's [7] idea of counting consistent cliques. A finer measure of similarity is provided by Sanfeliu and Fu's [1] idea using an edit-distance which counts node and edge relabelings together with the number of node and edge deletions or insertions necessary to transform one graph into another. More recently, in an attempt to cast the graph-matching problem into a Bayesian framework, Wilson and Hancock have shown in [8] how to construct a mixture model over a dictionary of structure-preserving mappings between the model graph and the data graph. Here, the distance between graphs depends on the Hamming distance between the node labels together with the size difference of the graphs. Although the dictionary can be compiled offline, its size can grow exponentially when the graphs are of different size and dummy nodes have to be inserted so as to model structural corruption due to the presence of clutter.

The aim in this paper is to focus more closely on the issue of how to measure the similarity of structurally corrupted graphs. The idea of using the Levenshtein or edit-distance to compare coded patterns which may have different sizes has existed for many years [9], [10]—indeed, the Hamming distance between two strings is a special case of the edit-distance. Wagner and Fischer used dynamic programming to evaluate the edit-distance between strings [10]. This idea has been extended to form a basis for comparing trees and graphs on a global level [11], [12], [1], [7]. More recently, the idea of actively editing graphs during the matching process to eliminate relational clutter has proved very successful [13], [8], [14]. The string edit-distance problem has received renewed interest in the pattern recognition literature [15], and Marzal and Vidal have recently shown how to normalize the edit-distance so that it may be consistently applied across a range of objects of different sizes [16]. Of particular relevance to the graph matching problem is the recent work of Messmer and Bunke [13] who exploited Sanfeliu and Fu's [1] graph-edit-distance to index multiple graph representations that have been encoded in a large model library using structural hashing. Finally, Bunke has recently demonstrated some interesting properties of the graph edit-distance. First, he has shown that the size of the maximum common subgraph is related to the edit-distance [17]. Second, he has commented on the uniqueness of the cost-function [18].

The observation underpinning this paper is that edit-distance represents an elegant alternative to the exhaustive compilation of dictionaries. Specifically, it provides a means by which structural errors can be modeled in an implicit rather than an explicit manner. Our goal is to follow Wilson and Hancock [19], [8] by modeling the probability distribution for edit-distance. We commence with a simple memoryless distribution rule over the basic edit operations. This leads to an exponential distribution. Although it can be shown that the dictionary-based graph-matching technique requires a polynomial number of dictionary comparisons, relatively little attention has been paid to the time and space complexity of dictionary compilation and lookup. In the original work on discrete relaxation, Waltz [20] had a large but fixed set of dictionaries for line labeling. Because it models structural error by padding out and permuting the nodes of graphs of different size, Wilson and Hancock's dictionary can grow exponentially. Although this growth can be curbed using relatively unobjectionable heuristics, the aim in this paper is to take a more principled approach. By adopting the edit-distance as our measure of similarity, we remove the need for dictionary padding and reduce the worst-case complexity to be polynomial. In an experimental study, we show that even a relatively naïve application of the edit-distance approach performs no worse than the original, and can do significantly better under certain circumstances.

The outline of this paper is as follows: In Section 2, we briefly review Wilson and Hancock's MAP framework for discrete relaxation. Section 3 describes the dictionary-based prior for graph-matching. In Section 4, we look critically at the complexity of dictionary compilation when there are structural errors and inexact graph-matching is being attempted. Section 5 introduces the edit-path concept and provides a Bayesian model of the associated prior. In Section 6, we provide some comparative experimental evaluation. This consists of both a sensitivity study and some real-world examples. Finally, Section 7 provides some conclusions and offers prospects for future work.

2 MAP Framework

We are interested in matching attributed relational graphs (ARGs) [1]. An ARG is a triple, $G = (V, E, A)$, where V is the set of vertices (nodes), E is the edge set ($E \subset V \times V$), and A is the set of node attributes ($A = \{\vec{x}_j | j \in V\}$). Consider a data graph $G_D = (V_D, E_D, A_D)$, which is to be matched onto a model graph $G_M = (V_M, E_M, A_M)$. The state of correspondence match can be represented by the function $f : V_D \mapsto V_M \cup \{\Phi\}$ from the node set of the data graph onto the node set of the model graph, where the node set of the model graph is augmented by adding a NULL label, Φ, to allow for unmatchable nodes in the

- *R.C. Wilson and E.R. Hancock are with the Department of Computer Science, University of York, United Kingdom. E-mail: {wilson, erh}@cs.york.ac.uk.*
- *R. Myers is with Praxis Critical Systems Limited, 20 Manvers St., Bath BA1 1PX, United Kingdom. E-mail: myers@praxis-cs.co.uk.*

Manuscript received 7 June 1999; revised 9 Mar. 2000; accepted 15 Mar. 2000.
Recommended for acceptance by S. Dickinson.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110009.

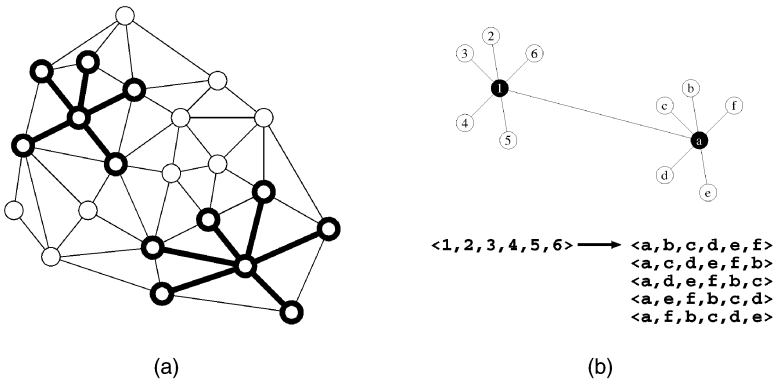


Fig. 1. Supercliques as defined by Wilson and Hancock in [8]. (a) Supercliques and (b) Possible mappings.

data graph. According to Wilson and Hancock's Bayesian framework [8], the quality of a match is measured by its a posteriori probability given the node attributes.

$$P(f|A_D, A_M) = \frac{p(A_D, A_M|f)}{p(A_D, A_M)} Q(f), \tag{1}$$

where $p(A_D, A_M|f)$ is the conditional measurement density, $p(A_D, A_M)$ is the joint measurement density, and $Q(f)$ is the matching prior. In [8], this criterion is optimized by hill climbing. That is, an initial match is established according to the measurement densities alone, and then iteratively updated so as to monotonically increase the value of the criterion. Assuming statistical independence of node attributes, the conditional measurement density, $p(A_D, A_M|f)$, can be factorized over the tuples (u, v) in the match f to yield an expression in terms of the a posteriori measurement probabilities, $P(u, v|\vec{x}_u, \vec{x}_v)$. Since the unconditional densities $p(A_D, A_M)$ and $p(\vec{x}_u, \vec{x}_v)$ are independent of the values of u and v, the criterion in (1) can be optimized by choosing a new value v for $f(u)$ at each iteration according to the following MAP update rule:

$$f(u) = \arg \max_{v \in V_M \cup \{\Phi\}} \frac{P(u, v|\vec{x}_u, \vec{x}_v)}{P(u, v)} Q(f). \tag{2}$$

The measurement densities are concerned with node attributes and are not our primary interest in this paper, although they are crucial ingredients of the overall matching strategy. We are concerned here with the matching prior, $Q(f)$. Wilson and Hancock average the matching prior, $Q(f)$, over the matching probabilities for the set of *supercliques* in the data graph. The superclique of the node i consists of its center node, together with its immediate neighbors connected by edges in the graph, i.e., $C_i^D = i \cup \{u; (i, u) \in E_D\}$. Supercliques are illustrated in Fig. 1a, which shows a graph with two of its supercliques highlighted. Since the neighborhoods or supercliques of neighboring nodes are overlapping, and their individual probabilities are hence not independent, Wilson and Hancock take a goal directed tack in averaging the matching prior over the data graph supercliques. The matching prior can be rewritten in terms of the probabilities of the images of the supercliques in the data graph under f:

$$Q(f) = \frac{1}{|V_D|} \sum_{i \in V_D} P(\Gamma_i), \tag{3}$$

where $\Gamma_i = (f(u_0), f(u_1), \ldots\ldots, f(u_{|C_i^D|})$ denotes the relational image of the superclique C_i^D in G_D under the matching function f.

3 Dictionary-Based Matching Prior

In this section, we review Wilson and Hancock's model of structural errors. This commences with a mixture model which computes the probability of the structure-preserving mapping Γ_i. The idea is to use the Bayes rule to expand the matching probability over a dictionary of legal structure-preserving mappings between the data and model graphs. The dictionary is compiled by considering the cyclic permutations of the peripheral nodes about the center node in the superclique, as shown in Fig. 1b. A complication arises from the fact that the supercliques being compared in the two graphs may be of different size due to clutter (i.e., noise) or dropout. Wilson and Hancock [8] dealt with this problem by adding padding to the smaller superclique and by screening out nodes from the larger superclique as appropriate. This is essentially a brute force method and may significantly add to the complexity of the dictionaries as we will show later.

Wilson and Hancock's [8] aim is to assign matches to the nodes in the data graph G_D by exploiting structural constraints provided by the supercliques of the model graph G_M. The constraints are represented by a dictionary of structure preserving mappings between the data graph superclique C_i^D and each of the supercliques C_j^M belonging to the model graph G_M. A series of entries are created in the dictionary Θ_i for the data graph node i for each of the model graph supercliques. If the model graph superclique is of the same size as the data graph superclique, then the entries are created by permuting the order of the peripheral model graph nodes about the center node. However, when the supercliques are of different size, then the smaller unit is padded with dummy nodes to raise it to the same size as the larger unit. This is a two step process. First, one or more dummy edges are inserted into the smaller superclique between each pair of the existing edges. Second, each of the resulting padded configurations undergoes cyclic permutation to generate dictionary entries. The process effectively models the disruption of the adjacency structure of the data graph caused by the addition of clutter elements or the loss of elements due to segmental dropout. It is important to stress that the center nodes are always paired with one another. The dictionary of feasible mappings generated in this way is denoted by $\Theta_i = \{S\}$. Each dictionary item is a structure-preserving mapping of the form

$$S = (s_0, s_1, .., s_u, ..., s_{|C_i^D|}), \tag{4}$$

where $s_u \in j \cup \{v; (j, v) \in C_j^M\} \cup \Phi$ is either one of the node labels drawn from the model graph superclique or the null label Φ, and $u \in C_i^D$ is one of the node labels drawn from the data graph superclique C_i^D.

With the dictionary at hand, Wilson and Hancock [8] use the Bayes formula to compute the matching probability, $P(\Gamma_i)$. This is done by expanding over the dictionary Θ_i in the following manner:

$$P(\Gamma_i) = \sum_{S \in \Theta_i} P(\Gamma_i|S).P(S). \tag{5}$$

The dictionary prior, $P(S)$, is generally taken to be uniformly distributed over the dictionaries, and, is hence, equal to $\frac{1}{|\Theta_i|}$. The conditional matching probability $P(\Gamma_i|S)$ is determined by comparing every assigned match $f(u)$ in the configuration Γ_i with the corresponding item s_u in the dictionary entry S. Wilson and Hancock use a model in which differences between the configuration and the dictionary item are the outcome of a memoryless label corruption process. Assuming statistical independence of these errors, they factorize over the superclique to give the configuration probability in terms of the atomic labeling probabilities

$$P(\Gamma_i|S) = \prod_{u \in C_i^D} P(f(u)|s_u) \tag{6}$$

and these atomic probabilities are, in turn, given by a simple distribution rule.

$$P(f(u)|s_u) = \begin{cases} P_e & \text{if } f(u) \neq s_u \\ (1 - P_e) & \text{otherwise.} \end{cases} \tag{7}$$

This gives an exponential form for $P(\Gamma_i)$ which depends on the label error probability, P_e, and the Hamming distance between the mappings and the dictionary, $\mathrm{H}(\Gamma_\mathrm{i}, \mathrm{S})$:

$$P(\Gamma_i) = \frac{K_{C_i^D}}{|\Theta_i|} \sum_{S \in \Theta_i} \exp[-k_e \mathrm{H}(\Gamma_\mathrm{i}, \mathrm{S})], \tag{8}$$

where $K_{C_i^D} = (1 - P_e)^{|C_i^D|}$, and $k_e = \ln \frac{(1-P_e)}{P_e}$.

The MAP estimate of the matching-function can be located by gradient ascent, following the framework set out in [8], or by global optimization techniques such as genetic search [14]. That is, an initial match is established according to the node attributes alone, and then, iteratively updated so as to monotonically increase the value of the criterion. The value of P_e is decreased from an initial high value to some arbitrarily small value, in a manner analogous to temperature change in an annealing process [21].

3.1 Dictionary Padding

It is usually the case that the sizes of the superclique C_i^D of the data graph node, i, and that of its relational image, $C_f(i)^M$, in the model graph, are different. In [8], Wilson and Hancock addressed this problem by padding the dictionary items with dummy labels so that it was the same size as the local configuration. This idea has been employed by several other authors including Wong and Ghahraman [22], Wong and You [23] and, more recently, by Sengupta and Boyer [24]. For example, consider the configuration, $\Gamma_i =< u_1, u_2, u_3 >$, and the dictionary item, $S =< v_1, v_2 >$. In order to compare the configuration to the dictionary item, padding must be added to the dictionary item to give $S' =< v_1, v_2, \Phi >$. If the dictionary item is larger than the configuration, the configuration must be padded. This approach to the problem entails several important drawbacks: First, an additional parameter is required; second, the number of such padded dictionary entries will be large, and third, summing over very many dictionary items (8) will distort the criterion.

To see the need for an extra parameter, consider the distribution rule in (7), in the case when dummy labels are present. Suppose that the consistent labeling, $< u_1, u_2, u_3 >$, has been corrupted by the addition of u_4 at the end. When comparing to the dictionary item $< v_1, v_2, \Phi, v_3 >$, one might conclude that the Hamming distance should be 2, whereas, in fact, only a single error has occurred. On the other hand, simply ignoring dummy labels would lead to considering $< u_1, u_2, u_3, u_4 >$ a perfect match for $< v_1, v_2, v_3, \Phi >$ even though there is an error in the labeling. To avoid these difficulties, Wilson and Hancock use the following distribution rule:

$$P(f(u)|s_u) = \begin{cases} P_\phi & \text{if } f(u) = \Phi \text{ or } s_u = \Phi \\ (1 - P_\phi)P_e & \text{if } f(u) \neq s_u \\ (1 - P_\phi)(1 - P_e) & \text{otherwise.} \end{cases} \tag{9}$$

The effect of this on the global criterion of (8) is to introduce as an additional control variable the probability of a label being added to or deleted from a configuration, P_ϕ. Wilson and Hancock found that explicitly controlling P_ϕ did not give particularly good results [8]. Whereas the introduction of P_e allowed the labeling process to be controlled in a principled manner, the introduction of P_ϕ only caused problems. A major part of Wilson and Hancock's work was the control of the process by which data graph nodes are assigned the NULL label. In [8], it was found that the best method for NULL labeling was graph editing, closely followed by a constraint filtering postprocessing step, both of which considerably outperformed explicit control of P_ϕ. However, that work did not address the consequences of dictionary padding for the space and time requirements of the evaluation of (8).

4 Complexity

The number of dictionary comparisons and, hence, the time complexity of the computation of $P(\Gamma_i)$ is clearly $\mathrm{O}(|\Theta_\mathrm{i}|.|\mathrm{S}|)$, and will depend on the sizes of the supercliques in the model graph and the amount of padding added to the dictionaries. The length of the structure-preserving mappings, $|S|$, is linear in the superclique size and the amount of padding, and will not play a significant role in the overall complexity. The dictionary Θ_i, however, is constructed by adding dummy items to the supercliques C_i^D in the model graph such that every possible combination is represented. If there are to be k NULLs, there are $|C_i^D| + k$ positions in the ith dictionary, of which k are NULL. Since the dictionary items are cyclic, only $|C_i^D| + k - 1$ of these positions are distinct. The order of the NULLs is not important, since they are indistinguishable from one another, so the number of possibilities is

$$\binom{|C_i^D| + k - 1}{k}.$$

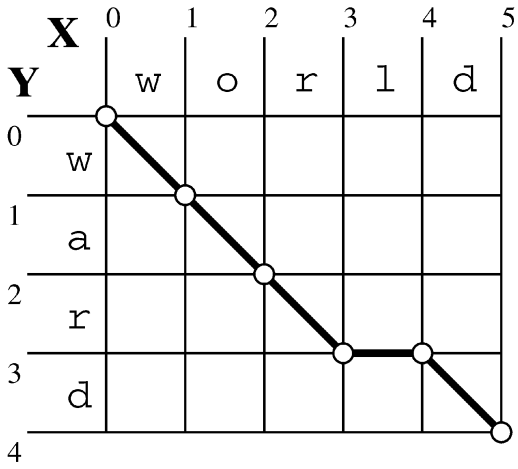
Since the mapping of the center node is fixed, for each value of k, the dictionary is composed of the $|C_i^D| + k - 1$ cyclic permutations of the augmented supercliques. Thus, the total size of the dictionary Θ_i is

$$|\Theta_i| = \sum_{0 \leq k < K_{\mathrm{MAX}}} (|C_i^D| + k - 1).\binom{|C_i^D| + k - 1}{k}. \tag{10}$$

This quantity is at a maximum when $k \approx |C_i^D|$. Hence, we can give an upper bound for the dictionary size in terms of the average superclique size $\overline{|C_i^D|}$.

$$\begin{aligned} |\Theta_i|_{\mathrm{MAX}} &= \mathrm{O}\left(V_M.2^{\overline{|C_i^D|}}.\frac{(2\overline{|C_i^D|})!}{\overline{|C_i^D|!}^2} \right) \\ &= \mathrm{O}\left(4^{\overline{|C_i^D|}}\right). \end{aligned} \tag{11}$$

It should be stressed that this is very much a worst-case scenario, which occurs infrequently. However, in the case of the two 50-node Delaunay graphs which we shall encounter in Section 6.4, the mean superclique size of the model graph is 5.5



(a)

$$\begin{aligned}
\Delta &= (\mathrm{w},\mathrm{w}),(\mathrm{o},\mathrm{a}),(\mathrm{r},\mathrm{r}),(\mathrm{l},\epsilon),\\
&\quad (\mathrm{d},\mathrm{d})\\
\mathbf{d} &= \gamma(\mathrm{o},\mathrm{a})+\gamma(\mathrm{l},\epsilon)=2\\
P &= (0,0),(1,1),(2,2),(3,3),\\
&\quad (4,3),(5,4)\\
W(P) &= 2\\
L(P) &= 6\\
\Rightarrow \hat{\mathbf{d}} &= \frac{1}{3}
\end{aligned}$$

(b)

Fig. 2. An example edit matrix from X to Y. In (a), the thick black line is the editing path. The relationship between the classical and normalized edit-distances is shown in (b).

(s.d. = 1.3) with a maximum of 9, and the mean size difference between superclique pairs from the two graphs is 1.4 (s.d. = 1.2), but has a maximum of 6. If we took the view that supercliques with size differences within 2 s.d.s of the mean should be considered, it would be necessary to add 4 NULLs to every superclique in order to form the dictionaries. According to (10), the estimated number of dictionary items that would have to be compiled is

$$\begin{aligned}
&50\sum_{0\le k<5}(6+k-1)\binom{6+k-1}{k}\\
&=50\times(5+6\times 6+7\times 21+8\times 56+9\times 126)\\
&=88,500,
\end{aligned}$$

which is considerable.

In practice, something like 80 percent of the pairs of supercliques have size differences of two or less. Given the underlying assumption of (7), that only one error occurs per mapping, the probability of significant superclique corruption is small. It is therefore reasonable to suppose that supercliques with size differences greater than 2 are not matchable anyway. In other words, setting the maximum number of NULLs at 2 gives manageable dictionaries and screens out some 20 percent of probably unmatchable superclique pairs, even though it violates the assumption of statistical independence that underpins (6). With only 2 NULLs, 9,400 dictionary items would be needed.

5 Edit Distance

Despite the success of the goal directed upper limit on dictionary padding in containing the underlying exponential time/space complexity of evaluating $Q(f)$, there remains a second theoretical weakness. The criterion relies on a rather artificial model in which dictionaries have to be padded so that they are the same size as the supercliques in the data graph. A measure of the distance between lists of differing lengths has existed for many years: the Levenshtein or string edit-distance [9], [10]. This avoids the use of padding altogether, by considering insertions and deletions in addition to changes. In what follows, we work with a simplified dictionary Θ_i^c which contains only cyclic permutations and whose size is therefore equal to $|C_i^D|-1$.

Let X and Y be two strings of symbols drawn from an alphabet Σ. We wish to convert X to Y via an ordered sequence of operations such that the cost associated with the sequence is minimal. The original string to string correction algorithm defined *elementary edit operations*, $(a,b)\neq(\epsilon,\epsilon)$, where a and b are symbols from the two strings or the NULL symbol, ϵ. Thus, changing symbol x to y is denoted (x,y), inserting y is denoted (ϵ,y), and deleting x is denoted (x,ϵ). A sequence of such operations which transforms X into Y is known as an *edit transformation* and denoted $\Delta=<\delta_1,\ldots,\delta_{|\Delta|}>$. Elementary costs are assigned by an elementary weighting function $\gamma:\Sigma\cup\{\epsilon\}\times\Sigma\cup\{\epsilon\}\mapsto\Re$; the cost of an edit transformation, $W(\Delta)$, is the sum of its elementary costs. The edit-distance between X and Y is defined as:

$$\mathbf{d}(X,Y)=\min\{W(\Delta)|\Delta \text{ transforms } X \text{ to } Y\}. \tag{12}$$

An interesting property of this quantity is that it is a metric if $\gamma>0$ for all nonidentical pairs and 0 otherwise, and if γ is selfinverse.

When used purely as a distance measure, the raw edit-distance, $\mathbf{d}$, may not be particularly useful for problems in which several comparisons must be made. If, for example, we wish to find the closest pair of strings in a set, as opposed to determining how to transform one string to another, correcting two errors between strings of size 3 should be more expensive than correcting five errors in strings of size 10. In [16], Marzal and Vidal address this problem by introducing the *normalized edit-distance*. They introduce the notion of an *edit path* which is a sequence of ordered pairs of positions in X and Y such that the path monotonically traverses the edit matrix of X and Y from $(0,0)$ to $(|X|,|Y|)$, as shown in Fig. 2. Essentially, the transition from one point in the path to the next is equivalent to an elementary edit operation: $(a,b)\rightarrow(a+1,b)$ corresponds to deletion of the symbol in X at position a. Similarly, $(a,b)\rightarrow(a,b+1)$ corresponds to insertion of the symbol at position b in Y. The transition $(a,b)\rightarrow(a+1,b+1)$ corresponds to a change from $X(a)$ to $Y(b)$. Thus, the cost of an edit path, $W(P)$, can be determined by summing the elementary weights of the edit operations implied by the path.

$$\mathbf{d}(X,Y)=\min\{W(P)|P \text{ is an edit path from } X \text{ to } Y\}. \tag{13}$$

According to Marzal and Vidal, this quantity can be normalized by dividing by the length of the path, $L(P)$. However, alternatives such as normalizing to the length of the shorter string are also possible. Thus, the normalized edit-distance is

$$\hat{\mathbf{d}}(X,Y) = \min\left\{\frac{W(P)}{L(P)} | P \text{ is an edit path from } X \text{ to } Y\right\} = \frac{W(P^*)}{L(P^*)}, \quad (14)$$

where P^* denotes the optimal path.

As Marzal and Vidal point out, it is clear from the form of $\hat{\mathbf{d}}$ that it cannot, in general, be determined by simply minimizing $W(P)$ and then dividing by $L(P)$. The complexity of computing $\hat{\mathbf{d}}$ by dynamic programming is $O(|X|.|Y|^2)$, $|X| \geq |Y|$ [16]; more efficient algorithms are also available [25], [26].

5.1 Application to Relational Matching

If we replace X and Y in Fig. 2 by a dictionary entry, S, and the image of a data graph superclique under the match, Γ_i, we can see that Γ_i could have arisen from S through the action of a memoryless error process, statistically independent of position (since the errors that "transformed" S into Γ_i could have occurred in any order). This means that we can still apply (7), except that we now factorize over the elementary operations implied by the transitions in P^* to give

$$P(\Gamma_i|S) = \prod_{(f(u),s_u) \Leftarrow P^*_{\Gamma_i,S}} P(f(u)|s_u), \quad (15)$$

where $(f(u), s_u)$ is an insertion, a deletion, a change or an identity operation implied by a transition in the minimum length edit path, $P^*_{\Gamma_i,S}$, between the superclique match, Γ_i, and the unpadded dictionary entry, S. An important feature of this approach is that the edit-distance, $\hat{\mathbf{d}}(\Gamma_i, S)$, is not used directly in the calculation of $P(\Gamma_i|S)$. The role of the edit-distance calculation is to obtain the sequence of edit operations which makes up the optimal path. Thus, the probabilities, $P(f(u)|s_u)$, need not necessarily be directly related to the edit weights, $\gamma(f(u), s_u)$. The edit weights will determine the optimal edit path of minimum cost, but it is the probabilities of the transitions in that path which contribute to the matching prior. When the different edit operations all have the same weights and probabilities, the maximum probability path has minimum edit-distance. If there is a more complicated distribution, then this is not necessarily the case. For this reason, we adopt a simplified distribution rule which corresponds to case in which the different edit operations have identical cost. The rule for assigning the probabilities to the edit operations is:

$$P(f(u)|s_u) = \begin{cases} (1-P_e) & \text{if } (f(u),s_u) \text{ is an identity} \\ P_e & \text{otherwise.} \end{cases} \quad (16)$$

The assumption of statistical independence also implies that the weighting function should be defined as follows:

$$\gamma(f(u),s_u) = \begin{cases} 0 & \text{if } (f(u),s_u) \text{ is an identity} \\ 1 & \text{otherwise.} \end{cases} \quad (17)$$

We can now write $P(\Gamma_i|S)$ as an exponential in terms of the number of nonidentity transformations in the optimal edit path from Γ_i to S. Given our weighting function in (17), this number is simply $W(P^*_{\Gamma_i,S})$, so the exponential of (8) becomes:

$$P(\Gamma_i) = \frac{1}{|\Theta_i^c|} \sum_{S \in \Theta_i^c} \exp\left[-\left(k_W W(P^*_{\Gamma_i,S}) + k_L L(P^*_{\Gamma_i,S})\right)\right], \quad (18)$$

where $k_W = \ln\frac{(1-P_e)}{P_e}$ and $k_L = \ln\frac{1}{(1-P_e)}$. If all edit operations have equal weights, the length of the optimal path will be equal to the length of the larger of $|\Gamma_i|$ and $|S|$.

The normalized edit-distance is not used directly. The criterion merely counts the elementary operations that make up the optimal path. The items in Θ_i no longer need to be padded, so using the edit-distance instead of dictionary padding reduces the worst-case space requirements of the dictionaries from $O(4^{|C_i^D|})$ to $O(|C_i^D|)$.

Although the edit-distance calculation is less efficient than a linear comparison with a padded dictionary, the number of dictionary comparisons required will be much less than with Wilson and Hancock's padded dictionary approach. Looking ahead to the example in Section 6.4, only 250 dictionary entries will be needed, regardless of the maximum tolerable size difference between matchable superclique. We should expect, therefore, that the edit-distance based method might be slower than the dictionary padding method when very small size differences are tolerated. As the maximum size difference is increased, the edit-distance method should eventually outperform the dictionary padding method.

In the special case studied here, where the editing operations are assumed statistically independent and the different types of edit operations are assigned identical probabilities, the normalized edit-distance can in fact be computed by postnormalization. The reason for this is that when these restrictions apply, the optimal path is the one with the minimum number of edit operations. Moreover, the postnormalized distance is efficiently computed by Wagner and Fischer's algorithm in $O(|X|.|Y|)$ time. Hence, by adopting the model of identical edit probabilities in (16), we are able to use the Wagner and Fisher method to compute the edit-distance and, hence, $P(\Gamma_i)$ efficiently.

5.2 Algorithm Complexity

In this section, we consider the time complexity involved in computing the probability of match $P(\Gamma_i)$ using the dictionary-based method in (7) and the edit-distance version appearing in (17). In each case, there are two contributions to the time complexity. The first of these is the complexity of computing the distance measure. The second is the number of distance computations that must be performed. The overall complexity is the product of these two contributions.

We commence by considering the case of the dictionary-based method. Suppose that C is the average superclique size. For exact matching, the dictionary consists of the cyclic permutations of those model graph supercliques which could match the ith data graph superclique, and the Hamming distance is computed in linear time, so the time taken is cubic in the configuration size and is $O(C^3)$. For inexact matching with padded dictionaries, the dictionary item comparison is still linear, but the worst-case dictionary size is as in (10), so here the time complexity is exponential in the average superclique size and is $O(C.4^C)$. If the normalized edit-distance is used, instead of the Hamming distance, as the distance measure, the distance evaluation is cubic but the dictionary size is now only quadratic, giving a quintic kernel, $O(C^5)$. Worst case complexities for these three cases are summarized in Table 1. The worst-case for exact matching is a quintic algorithm which is feasible. The worst-case for inexact matching with dictionary padding is infeasible. The worst-case for inexact matching with edit-distance may or may not be feasible depending on how powerful the hardware is, and how large C is.

It should be stressed that for exact matching and inexact matching with edit-distance, these worst-cases are also average cases. The average case for inexact matching with dictionary padding depends on the amount of padding.

6 Experiments

We have compared the new edit-distance method with the dictionary-based graph matching algorithm of Wilson and Hancock [8]. There are three aspects to this experimental evaluation. We commence with a sensitivity study where we investigate the effects of controlled relational corruption and initialization error in

TABLE 1
Worst Case Complexities

	Complexity		
Matching	Dictionary Size	Distance computation	Overall
Exact	$O(C^2)$	$O(C)$	$O(\|V_D\|\|V_M\|C^3)$
Inexact (padding)	$O(4^C)$	$O(C)$	$O(\|V_D\|\|V_M\|C4^C)$
Inexact (edit)	$O(C^2)$	$O(C^3)$	$O(\|V_D\|\|V_M\|C^5)$

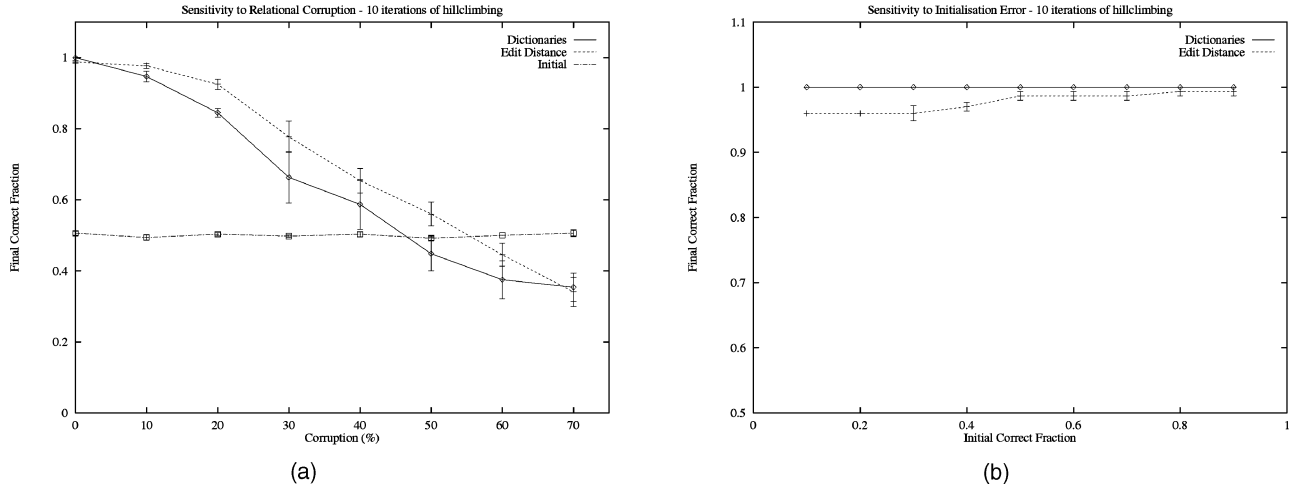


Fig. 3. Results of the sensitivity analysis. (a) Corruption and (b) initialization error.

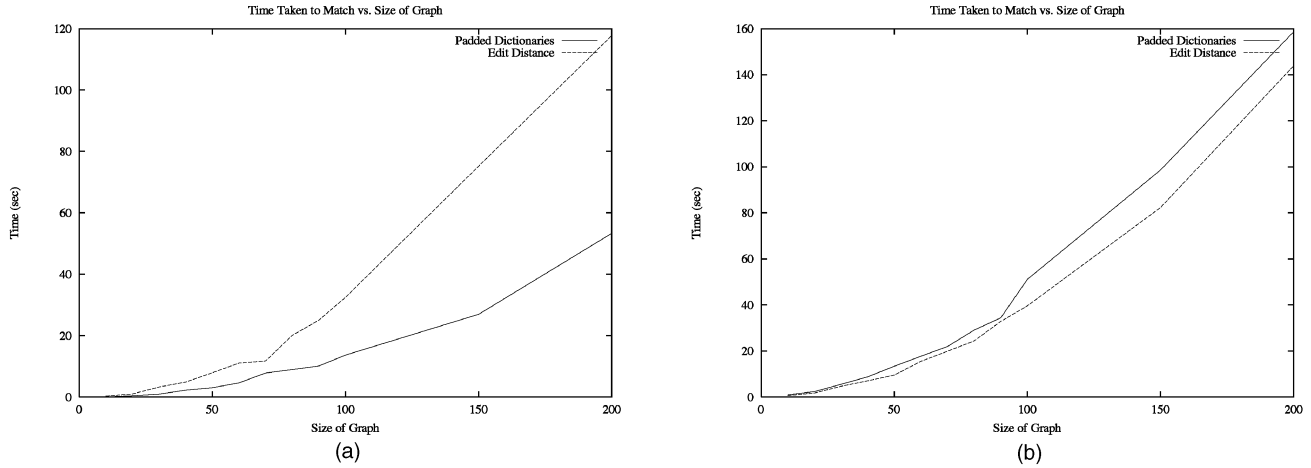


Fig. 4. Time required for evaluation of the functionals. Times for (b) (size difference ≤ 2) should generally be greater than for (a) (size difference ≤ 1) because there are more comparisons to be made.

a simulation study. The second aspect of our study assesses the efficiency of the two approaches. Finally, we consider some real world data, and demonstrate the effectiveness of the two algorithms on matching uncalibrated stereo images. Here, our work is addressing the structural stereo matching problem first tackled by Boyer and Kak [2].

6.1 Sensitivity Analysis

We created synthetic 50-node nearest-neighbor graphs with node attributes drawn from a uniform distribution. To simulate the effects of errors in feature detection, we randomly added and deleted nodes from the graphs and recomputed the nearest-neighbors. The relational disruption thus caused is greater in nearest-neighbor graphs than in the Delaunay triangulations used in [8]. We added Gaussian noise to the node attributes until approximately 50 percent of the nodes would be misclassified by a simple greedy classifier. The results of matching for both methods are shown in Fig. 3a. We determined the sensitivity to initialization error by adding varying amounts of Gaussian noise to the node attributes of uncorrupted graphs. The results are given in Fig. 3b.

6.2 Timing

To examine the efficiency of the two algorithms, we evaluated random matches between graphs of differing sizes. We have already established the theoretical complexity of the functionals in Sections 4 and 5. The experiments were done with the maximum

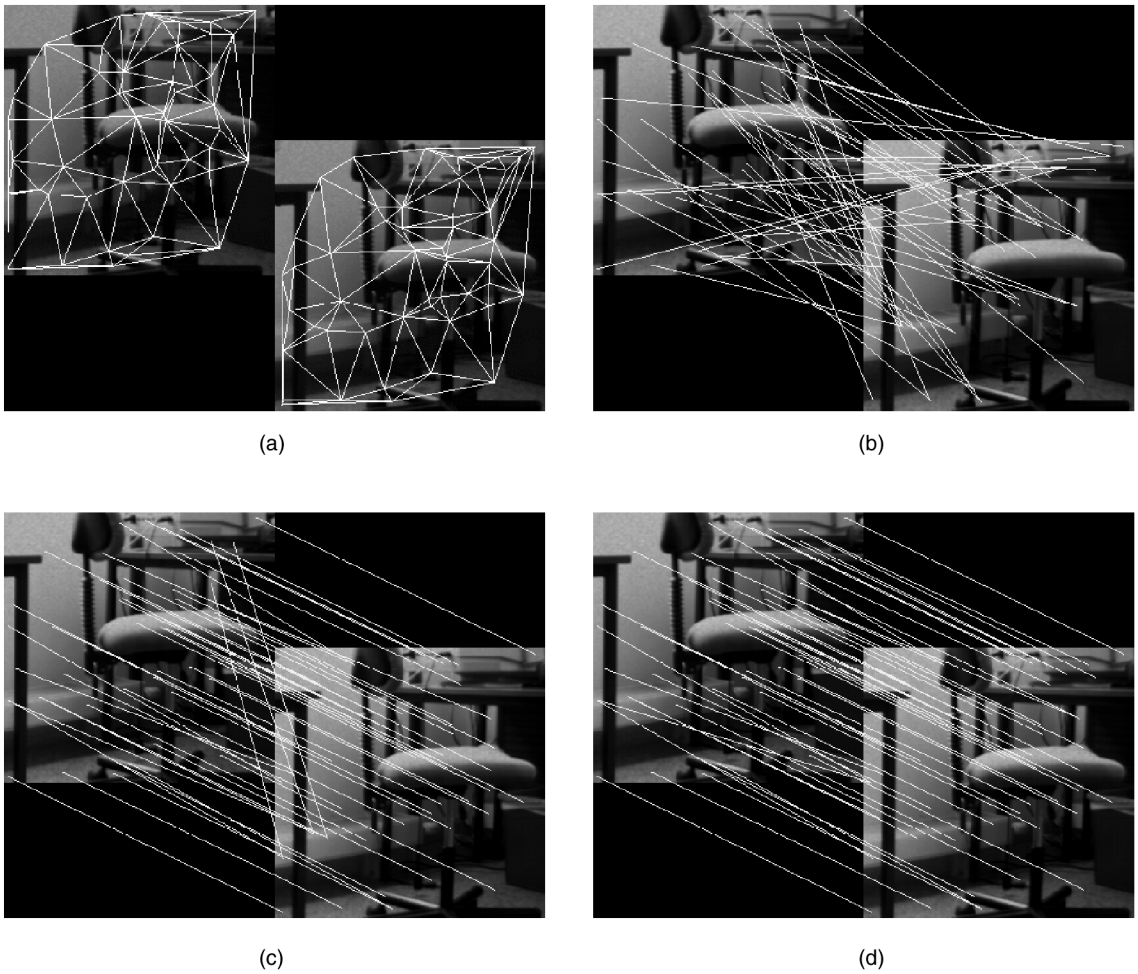


Fig. 5. Performance on uncalibrated stereo matching. Both algorithms restore a large proportion of the initial mappings despite about 35 percent relational corruption. The edit-distance method performs slightly better than the padded dictionary method. (a) Uncalibrated stereo pair, (b) initial guess, (c) final match (padded dictionaries), and (d) final match (edit-distance).

permitted superclique size difference set to either 1 or 2. There are a number of conclusions that can be drawn from the timing data shown in Fig. 4. First, it shows that for size differences less than or equal to one, then the dictionary-based method is more efficient than the edit-distance method. Second, it shows that when the size difference is less than or equal to two, then the two methods offer comparable performance. When the size difference is greater than two, then the edit-distance method is more efficient than the dictionary-based method.

6.3 Discussion

Fig. 3 shows that the edit-distance method comfortably outperforms the dictionary-padding techniques in the presence of relational corruption, but is more sensitive to initialization error. However, it should be remembered that the distribution rule in (16) is rather naïve in that it fails to penalize NULL labels even when the graphs have the same numbers of nodes. Moreover, all error types are treated as equivalent. This may explain the suboptimal performance observed for uncorrupted graphs. When this naïve distribution rule is applied for the original criterion in (8), the criterion performs much worse, unable to recover the correct match for uncorrupted graphs, and falling below the initial correct fraction for graphs with only 40 percent corruption.

This choice of distribution rule may also account for the poorer tolerance of the edit-distance criterion with respect to initialization error. With the Wilson and Hancock criterion, supercliques of differing cardinalities are not considered for matching when the graphs are of the same size, but with the edit-distance based criterion all supercliques are considered matchable. Artificially pruning the configuration space for matching improves the edit-distance based criterion's performance on initialization error.

6.4 Uncalibrated Stereo Matching

We demonstrate the practical applicability of the method on a wide baseline uncalibrated stereo matching problem. The lack of camera calibration makes this more difficult than the standard stereo correspondence problem. Regions were extracted from a gray-scale image pair (an office scene taken with an IndyCam) using a simple thresholding technique. Each image contained 50 regions. The region centroids were Delaunay triangulated using Triangle [27]. We used the average gray level over each region for the attribute information. The Delaunay triangulations were matched using a genetic algorithm with a local search step, starting from a random initial population. The maximum tolerated superclique size difference was two. Results are given in Fig. 5, which shows the two images with their Delaunay triangulations, the initial guess, and the results obtained when matching is effected using the padded dictionary and edit-distance methods. There were 50 regions in the left image of which 42 had feasible correspondences in the right. The initial guess contained no correct assignments. The padded dictionary method found 37 correct matches (88 percent), and the edit-distance method found 39 (93 percent). The amount of relational corruption between the two triangulations was estimated at around 35 percent by counting the number of inconsistent

supercliques given the ground truth match. A comparison of these results with Fig. 3a suggests that both methods are performing as expected from the synthetic data.

7 Conclusion

In this paper, we have shown that reformulating graph matching in terms of the edit-distance between supercliques remedies the worst-case exponential complexity of Wilson and Hancock's previous formulation [8]. For synthetic relational graphs, the new formulation gives an improvement of approximately 10 percent in matching accuracy in the presence of noise. When matches between supercliques with size differences greater than one are to be considered, the edit-distance method is more efficient than Wilson and Hancock's original.

Another conclusion to be drawn from this paper is that there is a taxonomy of graph matching problems which require different methods of making the dictionary comparison. For exact problems, in which the sizes of the supercliques and the dictionary items are the same, the Hamming distance suffices. For inexact problems where the error process is assumed to be memoryless, the postnormalized edit-distance must be calculated. Where the assumption of statistical independence must be dropped, the normalized edit-distance should be used. In this last case, the weighting function would also have to be inferred, perhaps by using a Markov model.

There are a number of ways in which the ideas presented in this paper can be extended. Although we have focussed on graph matching in this paper, the idea of using edit-distance is applicable to a wide range of consistent labeling problems. For instance, in a recent study, we have used the edit-distance prior for a simple natural language processing problem using tree-adjoining grammars [28]. One of our motivations for developing an efficient means of computing the prior was to provide us with a suitable vehicle for exploring consistent labeling problems using evolutionary optimization [29].

References

[1] A. Sanfeliu and K.S. Fu, "A Distance Measure Between Attributed Relational Graphs for Pattern Recognition," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 13, pp. 353-362, 1983.

[2] K.W. Boyer and A.C. Kak, "Structural Stereopsis for 3D Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, pp. 144-166, 1988.

[3] R. Horaud and T. Skordas, "Stereo Correspondence through Feature Grouping and Maximal Cliques," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, pp. 1,168-1,180, 1989.

[4] Y.C. Tang and C.S.G. Lee, "A Geometric Feature Relation Graph Formalism for Consistent Sensor Fusion," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 22, pp. 115-129, 1992.

[5] S.J. Dickinson, A.P. Pentland, and A. Rosenfeld, "3D Shape Recovery Using Distributed Aspect Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, pp. 174-198, 1992.

[6] H.G. Barrow and R.J. Popplestone, "Relational Descriptions in Picture Processing," *Machine Intelligence,* B. Meltzer and D. Michie, eds., vol. 6, Edinburgh Univ. Press, 1971.

[7] L. G. Shapiro and R.M. Haralick, "A Metric for Comparing Relational Descriptions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, pp. 90-94, 1985.

[8] R.C. Wilson and E.R. Hancock, "Structural Matching by Discrete Relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 634-648, 1997.

[9] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics-Doklady,* vol. 10, pp. 707-710, 1966.

[10] R.A. Wagner and M.J. Fischer, "The String-to-String Correction Problem," *J. ACM,* vol. 21, pp. 168-173, 1974.

[11] D. Shasha and K. Zhang, "Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems," *SIAM J. Computing,* vol. 18, pp. 1,245-1,262, 1989.

[12] K. Zhang, "A Constrained Edit Distance Between Unordered Labeled Trees," *Algorithmica,* vol. 15, pp. 205-222, 1996.

[13] B.T. Messmer and H. Bunke, "Efficient Error-Tolerant Subgraph Isomorphism Detection," *Shape, Structure, and Pattern Recognition,* D. Dori and A. Bruckstein, eds., World Scientific, 1994.

[14] A.D.J. Cross, R.C. Wilson, and E.R. Hancock, "Inexact Graph Matching Using Genetic Search," *Pattern Recognition,* vol. 30, pp. 953-970, 1997.

[15] H. Bunke and J. Csirik, "Parametric String Edit Distance and Its Application to Pattern Recognition," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 25, pp. 202-206, 1995.

[16] A. Marzal and E. Vidal, "Computation of Normalized Edit Distance and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 926-932, 1993.

[17] H. Bunke, "On a Relation Between Graph Edit Distance and Maximum Common Subgraph," *Pattern Recognition Letters,* vol. 18, pp. 689-694, 1997.

[18] H. Bunke, "Error Correcting Graph Matching: On the Influence of the Underlying Cost Function," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 917-922, 1999.

[19] R.C. Wilson, "Inexact Graph Matching Using Symbolic Constraints," PhD thesis, Department of Computer Science, Univ. of York, 1995.

[20] D. Waltz, "Understanding Line Drawings of Scenes with Shadows," *The Psychology of Computer Vision,* P.H. Winston, ed., pp. 19-91, McGraw-Hill, 1975.

[21] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, pp. 721-741, 1984.

[22] A.K.C. Wong and D.E. Ghahraman, "Random Graphs: Structural-Contextual Dichotomy," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, pp. 341-348, 1980.

[23] A.K.C. Wong and M. You, "Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, pp. 599-609, 1985.

[24] K. Sengupta and K.L. Boyer, "Organizing Large Structural Modelbases," *IEEE Trans. Pattern Analysis and Machine Intelligence,"* vol. 17, pp. 321-332, 1995.

[25] E. Vidal, A. Marzal, and P. Aibar, "Fast Computation of Normalized Edit Distances," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, pp. 899-902, 1995.

[26] B.J. Oommen and K. Zhang, "The Normalized String Editing Problem Revisited," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 669-672, 1996.

[27] J.R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," *Proc. Workshop Applied Computational Geometry,* pp. 124-133, 1996.

[28] R. Myers and E.R. Hancock, "Genetic Algorithms for Structural Editing," *Lecture Notes in Computer Science,* vol. 1,451, pp. 159-168, 1998.

[29] R. Myers and E.R. Hancock, "Genetic Algorithm Parameters for Line Labeling, *Pattern Recognition Letters,* vol. 18, pp. 1,363-1,371, 1997.