

# A Study of Graph Spectra for Comparing Graphs and Trees

Richard C. Wilson and Ping Zhu

*Computer Science Department  
University of York, UK  
wilson@cs.york.ac.uk*

---

## Abstract

The spectrum of a graph has been widely used in graph theory to characterise the properties of a graph and extract information from its structure. It has also been employed as a graph representation for pattern matching since it is invariant to the labelling of the graph. There are however a number of potential drawbacks in using the spectrum as a representation of a graph; Firstly, more than one graph may share the same spectrum. It is well known, for example, that very few trees can be uniquely specified by their spectrum. Secondly, the spectrum may change dramatically with a small change structure.

There are a wide variety of graph matrix representations from which the spectrum can be extracted. Among these are the adjacency matrix, combinatorial Laplacian, normalised Laplacian and unsigned Laplacian. Spectra can also be derived from the heat kernel matrix and path length distribution matrix. The choice of matrix representation clearly has a large effect on the suitability of spectrum in a number of pattern recognition tasks.

In this paper we investigate the performance of the spectra as a graph representation in a variety of situations. Firstly, we investigate the cospectrality of the various matrix representations over large graph and tree sets, extending the work of previous authors. We then show that the Euclidean distance between spectra tracks the edit distance between graphs over a wide range of edit costs, and we analyse the accuracy of this relationship. We then use the spectra to both cluster and classify the graphs and demonstrate the effect of the graph matrix formulation on error rates. These results are produced using both synthetic graphs and trees and graphs derived from shape and image data.

*Key words:* graph matching, tree matching, shape representation, spectrum, features

---

## 1 Introduction

The **spectrum of a graph** has been widely used in graph theory to characterise the properties of a graph and extract information from its structure. It has also been employed as a **graph representation for pattern matching tasks** [19,6,30]. Its use has not gained wide acceptance as a representation for matching and comparison of graphs. There are two main reasons for this, firstly, more than one graph may share the same spectrum. Secondly, the spectrum may change dramatically with a small change structure. While these factors appear to count against the spectrum, they may or may not be important in practical graph matching problems.

Graph structures have been used to represent structural and relational arrangements of entities in many vision problems. Some of the earliest attempts to do so are due to Fischler and Elschlager[7], and Barrow and Burstall[2]. More recently for example, shock graphs have been used to represent shape[20]. The key problem in utilising graph representations lies in measuring their structural similarity. This is a difficult problem because there is no explicit labelling of the parts, and typically correspondences must be established before similarity can be assessed. There are many methods in the literature which examine the problem of finding correspondences between graphs[25,9,24]. As an example, Sanfeliu and Fu[15] employed the concept of graph edit distance, giving separate edit costs for relabeling, insertion and deletion on both nodes and edges. A search is necessary to locate the set of operations which have minimal cost. More recently, Bunke[3,4] has established a relationship between the minimum graph edit distance and the size of the maximum common subgraph. The graph edit distance therefore provides a well-defined way of measuring the similarity of two graphs.

Spectral graph theory provides another approach to the problem of graph similarity. This approach is based on a branch of mathematics that is concerned with characterising the structural properties of graphs using the eigenvectors and eigenvalues of the adjacency matrix or the closely related Laplacian matrix (the degree matrix minus the adjacency matrix) [5]. **One of the well known successes of spectral graph theory in computer vision is the use of eigenvector methods for grouping via pairwise clustering.** Examples include Shi and Malik's [18] iterative normalised cut method which uses the Fiedler (i.e. second smallest) eigenvector for image segmentation and Sarkar and Boyer's use of the leading eigenvector of the weighted adjacency matrix [16]. Graph spectral methods have also been used for correspondence analysis. For example, **Umeyama's method[22] allows the matching of two graphs of equal size by using the eigendecompositions of the adjacency matrices.** Kosinov and Caelli[11] have used properties of the spectral decomposition to represent graphs and **Shokoufandeh et al[19] have used eigenvalues of shock graphs to index shapes.**

We have previously shown[26,27] how permutation invariant polynomials can be used to derive features which describe graphs and make full use of the available spectral information.

The spectrum of a graph (i.e. the set of eigenvalues) is generally considered to be too weak to be a useful tool for representing graphs, mainly due to the result of Schwenk[17] who showed that for trees at least, a sufficiently large tree nearly always has a partner with the same spectrum. **Trees therefore cannot be uniquely defined by the spectrum.** However, it is not known to what extent this is a problem in practice since Schwenk’s work does not reveal how large ‘sufficiently large’ actually is. The situation for graphs is even less clear, as no similar result is known. Computational simulations by Haemers et al[23] have shown that the fraction of cospectral graphs reaches 21% at 10 vertices (for the adjacency matrix) and is less for 11 vertices, which is the limit of their simulations. While far from conclusive, their results suggest that for small graphs, the fraction of cospectral graphs is small for some representations, at least at 11 vertices, the trend is decreasing.

The graph spectrum is derived from a matrix representation of the graph, and is highly dependent on the form of the matrix. A number of alternative matrix representations for graphs have been proposed in the literature; these include the **adjacency matrix, Laplacian and normalised Laplacian**. The spectrum of all of these representations may be used to characterise the graph, and each may reveal different graph properties. Some of these representations may be more stable to perturbations in the graph. In this paper we analyse these matrices and quantify the effect the matrix representation has on the stability and representational power of the eigenvalues of the graph. We also examine the problem of cospectrality of graphs. In section 2, we discuss the spectral decomposition of a graph. Section 3 describes the standard matrix representations of graphs. In section 4, we investigate the cospectrality properties of these matrices with respect to trees and general graphs. In Section 5 we look at the relationship between graph spectra and the edit distance between graphs. Finally, section 6 details the experiments aimed at measuring the utility of these representations in more practical situations such as clustering and classification.

## 2 Spectral decomposition of the representation matrix

**The graphs under consideration here are undirected, unweighted graphs.** While we do not consider weighted graphs here, these ideas are straightforwardly extended to such graphs. We denote a graph by  $G = (V, E)$  where  $V$  is the set of nodes and  $E \subseteq V \times V$  is the set of edges. The degree of a vertex  $u$  is the number of edges incident on the vertex  $u$  and is denoted  $d_u$ . A *matrix*



Fig. 1. A pair of graphs with the same adjacency matrix spectrum[10]

*representation* of the graph is a  $|V|$  by  $|V|$  matrix  $X$ , such that an element  $X_{ij}$  of this matrix represents some property of the pair of vertices  $i$  and  $j$ . Diagonal elements  $X_{ii}$  encode information about the vertex  $i$  only. A simple example is the adjacency matrix  $A$ , where  $A_{ij}$  is 1 when there is an edge between  $i$  and  $j$ , and zero otherwise. We discuss specific representations in more detail in the next section.

The spectrum of the graph is obtained from the matrix representation using the eigendecomposition. Let  $\mathbf{X}$  be the matrix representation in question. Then the eigendecomposition is  $\mathbf{X} = \Phi \Lambda \Phi^T$  where  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{|V|})$  is the diagonal matrix with the ordered eigenvalues as elements (ordered in terms of magnitude, with the largest first) and  $\Phi = (\phi_1 | \phi_2 | \dots | \phi_{|V|})$  is the matrix with the ordered eigenvectors as columns. The spectrum is the set of eigenvalues

$$s = \{\lambda_1, \lambda_2, \dots, \lambda_{|V|}\}$$

with

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{|V|}$$

One of the key problems with graph comparison is that the vertices are not labelled or ordered. As a result, the vertices may appear in different orders in two graphs under comparison, even if the graphs are isomorphic. Usually, correspondences between the vertices in the two graphs must be established before similarity can be measured. Clearly, the vertex ordering affects the matrix representation of the graph. If  $\mathbf{P}$  is a permutation matrix which re-orders the vertices, then

$$\mathbf{L}' = \mathbf{P} \mathbf{L} \mathbf{P}^T$$

represents the same graph as  $\mathbf{L}$ .

The spectrum is particularly useful as a graph representation because it is invariant under the similarity transform  $\mathbf{P} \mathbf{L} \mathbf{P}^T$ . In other words, two isomorphic graphs will have the same spectrum. This is the motivation for using the spectrum as a graph feature. As noted earlier, the converse is not true, two nonisomorphic graphs may share the same spectrum. Figure 1 shows an example of two such graphs which have the same adjacency matrix spectrum from [10]. However, the spectrum may be used as an approximate measure of graph similarity. One of the aims of this paper is to establish how useful the spectrum is as such a measure.

The spectral distance between graphs is simply the Euclidean distance between

the spectra.

$$d_s(G_1, G_2) = \sqrt{\sum_i (s_i^{(1)} - s_i^{(2)})^2}$$

When the spectra are of different sizes, then the smaller may be padded with zero values (while maintaining the magnitude ordering). This is equivalent to adding disjoint vertices to the smaller graph to make both graphs the same cardinality.

### 3 Standard Graph Representations

In this section, we review the properties of some standard graph representations and their relationships with each other.

#### 3.1 Adjacency matrix

The most basic matrix representation of a graph is using the adjacency matrix  $\mathbf{A}$  for the graph. This matrix is given by

$$A(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Clearly if the graph is undirected, the matrix  $\mathbf{A}$  is symmetric. As a consequence, the eigenvalues of  $\mathbf{A}$  are real. These eigenvalues may be positive, negative or zero and the sum of the eigenvalues is zero. The eigenvalues may be ordered by their magnitude and collected into a vector which describes the graph spectrum.

#### 3.2 Combinatorial Laplacian matrix

In some applications, it is useful to have a positive semidefinite matrix representation of the graph. This may be achieved by using the Laplacian. We first construct the diagonal degree matrix  $\mathbf{D}$ , whose diagonal elements are given by the node degrees  $D(u, u) = d_u$ . From the degree matrix and the adjacency matrix we then can construct the standard combinatorial Laplacian matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (2)$$

i.e. the degree matrix minus the adjacency matrix. The Laplacian has at least one zero eigenvalue, and the number of such eigenvalues is equal to the number

of disjoint parts in the graph. This matrix represents a discrete version of the Laplacian in continuous space.

The *signless* Laplacian has all entries greater than zero and is defined to be

$$|\mathbf{L}| = \mathbf{D} + \mathbf{A} \quad (3)$$

This matrix has been studied previously in [10,23] and seems to have good properties in the sense that it produces fewer cospectral graphs than the Laplacian, at least for small graphs[10].

### 3.3 Normalized Laplacian matrix

The *normalized Laplacian matrix* is defined to be the matrix

$$\hat{L}_{uv} = \begin{cases} 1 & \text{if } u = v \\ -\frac{1}{\sqrt{d_u d_v}} & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

We can also write it as  $\hat{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$ . As with the Laplacian of the graph, this matrix is positive semidefinite and so has all eigenvalues greater than or equal to zero. The normalisation factor means that the largest eigenvalue less than or equal to 2, with equality only when  $G$  is bipartite. Again, the matrix has at least one zero eigenvalue. Hence all the eigenvalues are in the range  $0 \leq \lambda \leq 2$ .

This matrix is described in[5] and contains information about a random walk on the graph. It is commonly used as a graph representation and graph cuts[18,5,14].

### 3.4 Heat Kernel

The *heat kernel* is based on the diffusion of heat across the graph. It is closely related to the flow of information through the graph with time[29]. It is a representation which has attracted recent interest in the literature for graph applications such as clustering[1]. We are interested in the heat equation associated with the normalised Laplacian, i.e.  $\frac{\partial \mathbf{H}_t}{\partial t} = -\hat{\mathbf{L}} \mathbf{H}_t$  where  $\mathbf{H}_t$  is the heat kernel and  $t$  is time. The solution is found by exponentiating the normalised Laplacian eigenspectrum, i.e.  $\mathbf{H}_t = \Phi \exp[-t\Lambda] \Phi^T$ . The heat kernel is a  $|V| \times |V|$  matrix, and for the nodes  $u$  and  $v$  of the graph  $G$  the resulting

component is

$$H_t(u, v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t] \phi_i(u) \phi_i(v) \quad (5)$$

When  $t$  tends to zero, then  $\mathbf{H}_t \simeq \mathbf{I} - \hat{\mathbf{L}}t$ , i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand,  $t$  is large, then  $\mathbf{H}_t \simeq \exp[-t\lambda_m] \phi_m \phi_m^T$ , where  $\lambda_m$  is the smallest non-zero eigenvalue and  $\phi_m$  is the associated eigenvector, i.e. the **Fiedler vector**. Hence, the large time behavior is governed by the global structure of the graph. By controlling  $t$ , we can obtain representations of varying degrees of locality.

### 3.5 Path Length Distribution

It is interesting to note that the heat kernel is also related to the path length distribution on the graph. If  $D_k(u, v)$  is the number of paths of length  $k$  between nodes  $u$  and  $v$  then

$$H_t(u, v) = \exp[-t] \sum_{k=1}^{|V|^2} D_k(u, v) \frac{t^k}{k!} \quad (6)$$

The path length distribution is itself related to the eigenspectrum of the normalised Laplacian. By equating the derivatives of the spectral and the path-length forms of the heat kernel it is straightforward to show that

$$D_k(u, v) = \sum_{i=1}^{|V|} (1 - \lambda_i)^k \phi_i(u) \phi_i(v) \quad (7)$$

Hence,  $D_k(u, v)$  can be interpreted as the sum of weights of all walks of length  $k$  joining nodes  $u$  and  $v$ .

## 4 Cospectrality of graphs

Two graphs are said to be *cospectral* if they have the same eigenvalues with respect to the matrix representation being used. The spectrum of a graph has not, in the past, been considered as useful for comparing graphs, due to the result of Schwenk[17] who showed that for trees at least, a sufficiently large tree nearly always has a partner with the same spectrum. Trees therefore cannot be uniquely defined by the spectrum. This result does not, however, tell us

the size that trees need to be before this becomes a problem. The trees may need to be very large in order for there to be a significant chance of finding a cospectral pair. No similar result is known for general graphs, although there are subfamilies with cospectral graphs such as the set of strongly regular graphs. However, the matrix representation chosen for the graph has a large impact on the chance of finding cospectral graphs.

Haemers and Spence[10] have investigated the cospectrality of graphs up to size 11, extending a previous survey by Godsil and McKay[8]. Above 11 vertices it becomes computationally infeasible to enumerate and check the cospectrality of all graphs. In their study, they considered the adjacency matrix, the Laplacian and the signless Laplacian. We have extended this study by including results for the normalised Laplacian. These results are plotted in Figure 2, bottom right alongside the results of Haemers and Spence. They show that the adjacency matrix appears to be the worst representation in terms of producing a large number of cospectral graphs. The Laplacian is superior in this regard and the signless Laplacian even better. The normalised Laplacian initially appears inferior but the fraction of cospectral graphs rapidly falls and becomes far smaller than for the other representations for larger graphs. The normalised Laplacian produces approximately 0.2% cospectral graphs with 11 vertices. In contrast, the signless Laplacian, Laplacian and adjacency matrix produces 3.8%, 9% and 21% respectively. It is not clear why the normalised Laplacian is superior here, other than to note that the normalised Laplacian encodes information about the degrees of the vertex which adjoining an edge.

It is interesting to note that it may be possible to reduce the cospectrality problem further by combining more than one spectrum. We could, for example, compute the spectrum of the adjacency matrix and the Laplacian at the same time and consider graphs cospectral only if they are equal in both spectra. The value of this of course depends on whether the cospectral pairs of graphs are the same in each spectrum. Table 1 shows the results of combining spectra in this way for connected graphs. This results in a large reduction in the number of cospectral pairs, compared with the individual spectra, particularly for sizes less than 10 vertices. Combining the normalised Laplacian spectrum with another spectrum for 10 vertex graphs roughly halves the number of cospectral graphs. However, even combining all the representations does not significantly reduce this number; there is a core of slightly over 10000 graphs which are cospectral under all the representations.

As far as trees are concerned, we have complemented the investigation of Haemers and Spence by looking at the cospectrality of trees up to size 23 for all representations and size 26 for the Laplacian. These trees were generated using the method described by Li and Ruskey[12]. This method generates trees by recursively adding vertices, while maintaining a canonical labelling of the tree. This prevents regeneration of isomorphic copies of the tree. For trees,



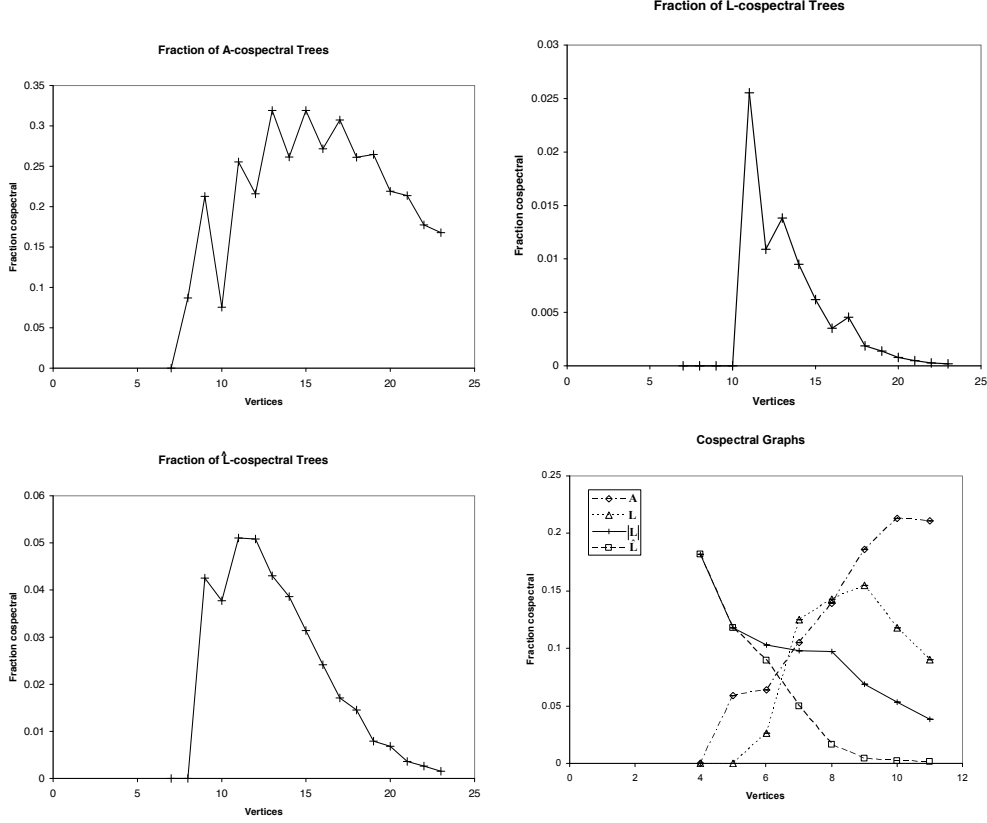


Fig. 2. Fractions of trees which are cospectral with respect to the matrices  $\mathbf{A}$ ,  $\mathbf{L}$  and  $\hat{\mathbf{L}}$ , and fractions of cospectral graphs[10]

Size	Number	$\mathbf{A}\&\mathbf{L}$	$\mathbf{L}\& \mathbf{L} $	$ \mathbf{L} \&\mathbf{A}$	$\hat{\mathbf{L}}\&\mathbf{A}$	$\hat{\mathbf{L}}\&\mathbf{L}$	All
6	112	0	0	0	0	0	0
7	853	0	16	0	0	0	0
8	11117	0	232	0	6	0	0
9	261080	82	4139	8	14	6	2
10	11716571	13864	107835	10716	10281	10256	10124

Table 1

Numbers of cospectral graphs when using spectra in combination (All: all representations are combined)

the spectrum of  $|\mathbf{L}|$  is directly related to  $\mathbf{L}$ , and the spectra of the heat kernel and path-length distribution can be derived from  $\hat{\mathbf{L}}$ . We therefore confine our attention to  $\mathbf{A}$ ,  $\mathbf{L}$  and  $\hat{\mathbf{L}}$ .

The results are summarised in Figure 2 and Table 2. The fractions here refer to the number of trees which do not have a unique spectrum. The Laplacian is clearly superior in this regard, having a very small fraction of cospectral graphs at all sizes. Both the Laplacian and its normalised counterpart show a decreasing trend, suggesting that for larger trees the fraction which are cospectral in these matrices could be negligible. The trend for the adjacency

Size	Number	$\mathbf{A}$	$\mathbf{L}$	$\hat{\mathbf{L}}$	$\mathbf{A} \text{ \& } \mathbf{L}$ (number)
8	23	0.087	0	0	0
9	47	0.213	0	0.0426	0
10	106	0.075	0	0.0377	0
11	235	0.255	0.0255	0.0511	2
12	551	0.216	0.0109	0.0508	2
13	1301	0.319	0.0138	0.0430	2
14	3159	0.261	0.0095	0.0386	10
15	7741	0.319	0.0062	0.0314	2
16	19320	0.272	0.0035	0.0241	14
17	48629	0.307	0.0045	0.0171	40
18	123867	0.261	0.0019	0.0145	38
19	317955	0.265	0.0014	0.0079	64
20	823065	0.219	0.0008	0.0068	148
21	2144505	0.213	0.0005	0.0036	134
22	5623756	0.177	0.00028	0.0026	134
23	14828074	0.168	0.00019	0.0015	134
24	39299897	-	0.00009	-	-
25	104636890	-	0.00007	-	-
26	279792891	-	0.00005	-	-

Table 2

Fractions of trees which are cospectral with respect to the matrices  $\mathbf{A}$ ,  $\mathbf{L}$  and  $\hat{\mathbf{L}}$

matrix is less clear, but the fraction appears to decrease after 15 vertices. Our results clearly show that the combinatorial Laplacian is by far the best representation in terms of the fraction of trees uniquely represented by the spectrum. This result is in line with that of Haemers and Spence[10] and our results for general graphs which suggested that the signless Laplacian was the best. These results are very encouraging since they suggest that for both trees and graphs, the cospectrality problem is negligible even for moderately sized objects.

The final column in Table 2 shows the number of pairs of trees which are cospectral in  $\mathbf{A}$  and  $\mathbf{L}$  at the same time. Interestingly, cospectral pairs for  $\mathbf{A}$  and  $\mathbf{L}$  seem to be unrelated to each other, and so combining the two spectra leads to very few cospectral graphs. For example, at 21 vertices, there are only 134 cospectral examples for  $\mathbf{A}$  and  $\mathbf{L}$  combined, from more than 2 million trees.

## 5 Measuring the stability and representational power of eigenvalues

One of the key questions which we aim to answer in this paper is how effective the spectrum of a graph is as a measure the similarity of two graphs to each other, and what impact does the representation have? Can the distance between spectra be used to measure the dissimilarity between graphs? Clearly this relationship cannot be precise since, as we have already seen, there are a number of graphs which share the same spectra but are not isomorphic. The relationship we are looking for is a statistical one; how well on average does the spectral distance reproduce the distance between graphs, and what is the variance in this relationship?

### 5.1 *Graphs of different sizes*

The spectrum of a graph is equal in size to the number of vertices in the graph. While this is not a problem when considering cospectrality since the graphs must be of the same size, in the more general case we must consider the possibility that the graphs are of different sizes and the spectra are of different lengths. The natural way to deal with this is to introduce extra vertices to the smaller graph to make it the same cardinality as the larger. These vertices have no edges associated with them and so the effect will be to add zeros to the spectrum of the smaller graph; the smaller spectrum is padded with zeros. Since we introduce new vertices without cost, this approach is essentially concerned with edge differences only.

### 5.2 *Graph distance*

There are a number of ways to measure the distance between two graphs, but the most appropriate in this case is the **edit distance**[15,3]. The edit distance is defined by a sequence of operations, including edge and vertex deletion and insertion, which transform one graph into another. Each of these operations has an associated cost, and the total cost of a sequence of edits is the sum of the individual costs. The sequence of minimal cost which transforms one graph into another is the edit distance between the graphs. For the spectral distance, we add vertices at no cost in order to make the graphs equal size. In order to be consistent with this, **we allow vertex insertions with zero cost**. We are concerned only with changes in the edge structure. Since the spectral distance is symmetric with regards to swapping the graphs, **we assign an equal cost of 1 for edge insertions and deletions**. Clearly, if the spectrum is to be a

good representation in this sense, then the requirement is that the distance between spectra should be related to the edit distance between the graphs.

In the experiments below we use edit distance on generated datasets as a comparison to the spectral distance. In these datasets we already have the true match between the graphs and so we already know the minimum edit distance and do not need to concern ourselves with locating it via relabelling operations.

### 5.3 Relationship between spectral distance and edit distance

In order to establish the relationship between edit distance and the spectrum, we begin with a set of randomly generated graphs. The graphs are of two sorts; the first is a Delaunay graph generated from a uniform random set of points, representing the type of graph derived from a image; the second is a more abstract graph formed by randomly connecting vertices (random graph). Both types of graph have 30 vertices and the random graph has 70 edges. The number of edges in the Delaunay graph is determined by the triangulation but is typically around 80.

The graphs are altered by deleting edges and then we compare the edit distance with the spectral distance, as measured by the Euclidean distance between the vectors of eigenvalues from the original and edited graphs. The edge to be deleted is chosen at random. For each level of editing, we perform 100 trials in order to obtain an average and deviation in the distance. The  $t$  in the heat kernel equation is set to 3.5 and the length of path in the path length distribution is 2. The results are shown in Figures 3 and 4.

These plots show that all these representations give a spectrum which follows the edit distance we have chosen closely, although the adjacency and Laplacian matrices seem marginally less linear. In Tables 3 and 4 we give the relative deviation of the samples for 5, 10, 20 and 30 edit operations. The relative deviation is the standard deviation of the samples divided by the mean. This value gives an indication of how reliably the spectrum predicts the edit distance. In this regard, the heat kernel matrix is clearly superior to the other methods.

## 6 Classification and Clustering

Classification and clustering are two central tasks in pattern recognition and they are of huge practical importance. In particular, similarity based classifi-

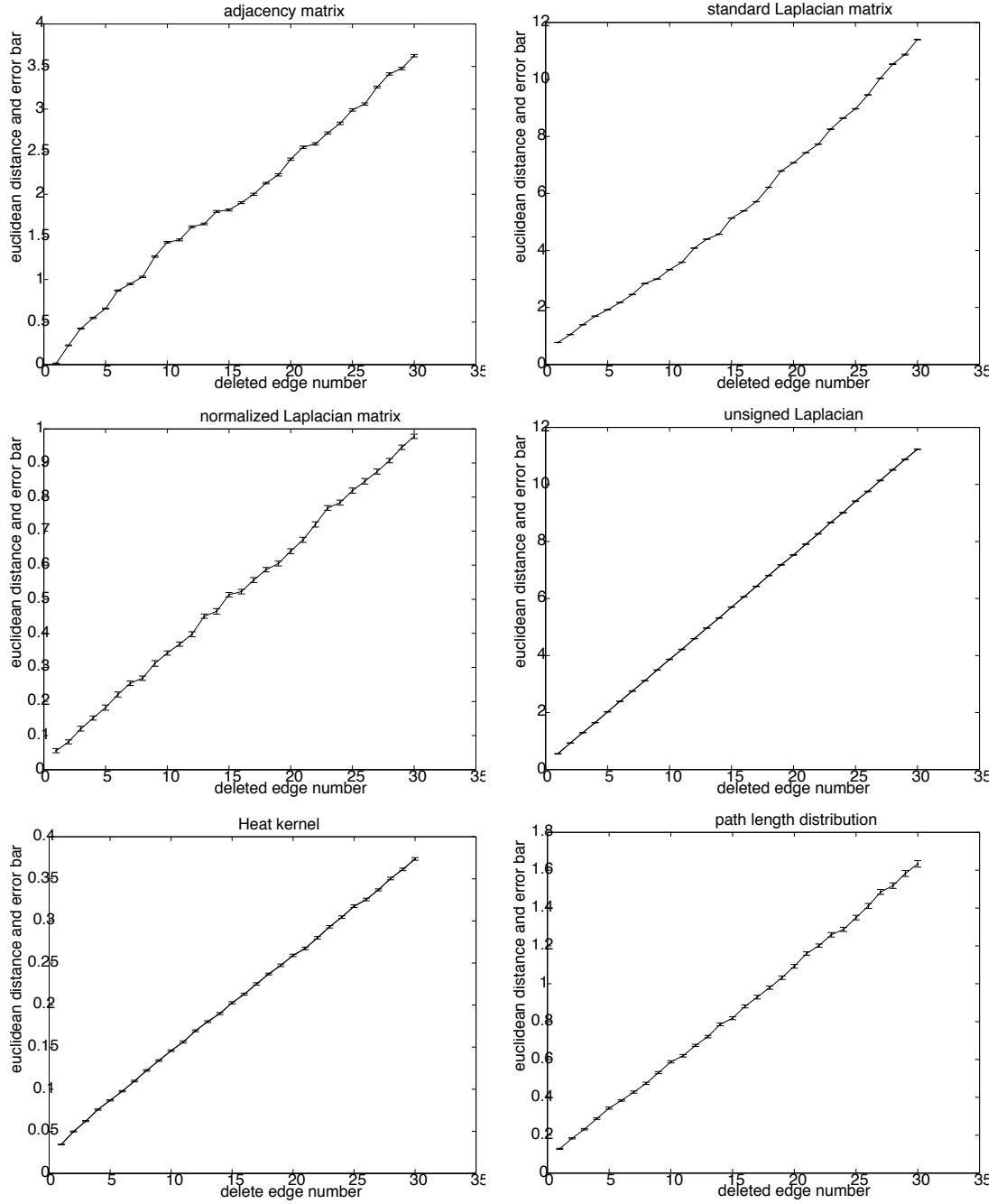


Fig. 3. Euclidean distance of Delaunay graphs

cation is an important task for graphs, particularly on large graph databases, where fast methods are required. The spectrum of the graph would be an ideal tool in such instances because it is easy to compute from the matrix representation of the graph. The performance of different matrix representations in this regard is therefore very interesting. If the spectrum of the matrix is a good representation, then we should be able to group similar graphs together and identify the class of a graph even under noisy conditions. In our second set of experiments, we therefore investigate the clustering and classification of

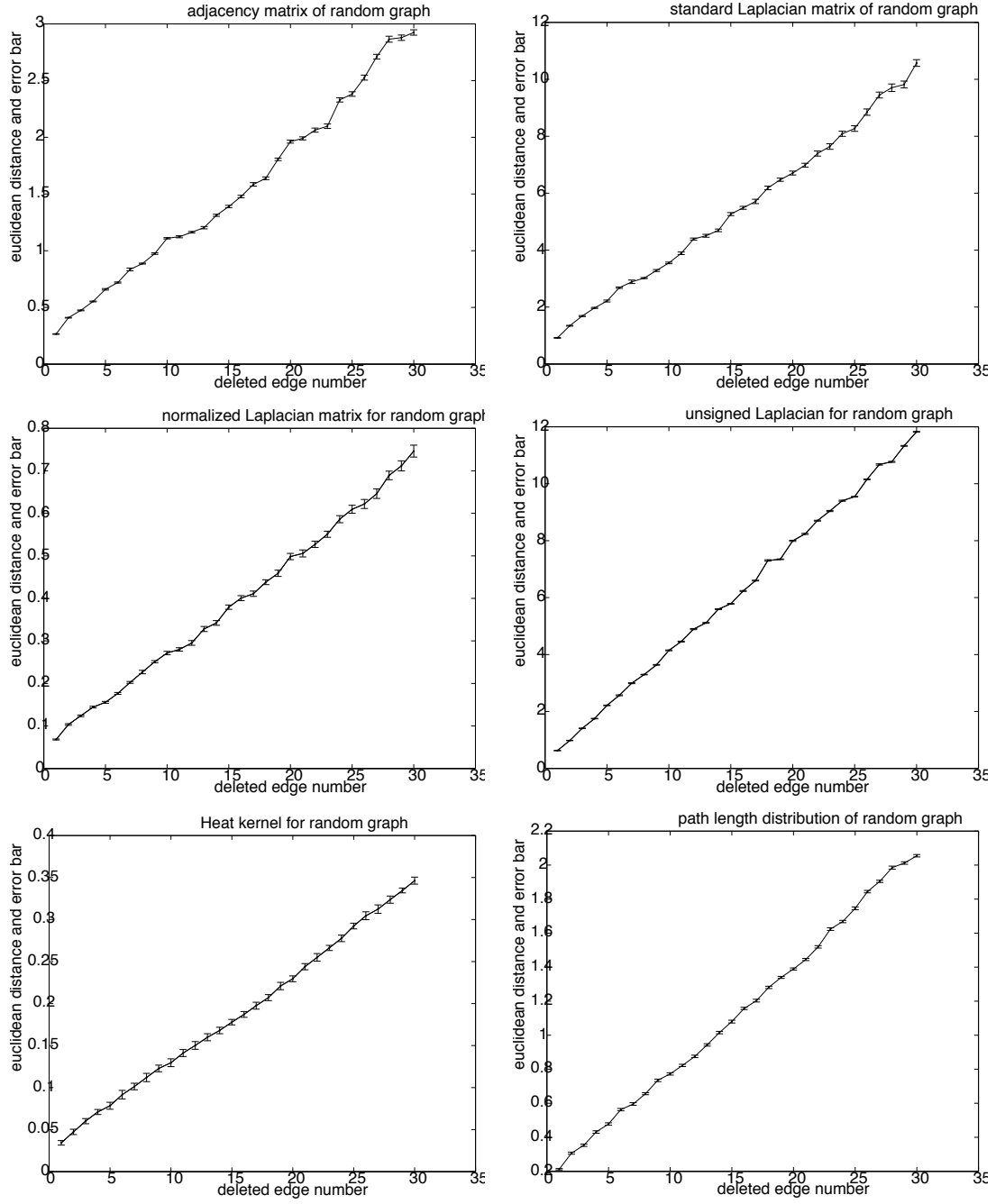


Fig. 4. Euclidean distance of random graphs

graphs on a number of different datasets.

### 6.1 Synthetic data

Our first dataset consists of random graphs. The graphs have 50 vertices and 200 randomly generated edges. The advantage of using this synthetic data is

Matrix	5 edge deletion	10 edge deletion	20 edge deletion	30 edge deletion
<b>A</b>	0.0918	0.0827	0.0716	0.0530
<b>L</b>	0.0802	0.0727	0.0619	0.0498
$\hat{\mathbf{L}}$	0.0753	0.0676	0.0571	0.0414
$ \mathbf{L} $	0.0523	0.0449	0.0268	0.0121
$\mathbf{H}_t$	0.0358	0.0287	0.0193	0.0105
$\mathbf{D}_k$	0.0420	0.0313	0.0252	0.0127

Table 3

Relative deviation of spectral distances for Delaunay graphs

Matrix	5 edge deletion	10 edge deletion	20 edge deletion	30 edge deletion
<b>A</b>	0.1164	0.1023	0.0805	0.0657
<b>L</b>	0.1042	0.0930	0.0771	0.0592
$\hat{\mathbf{L}}$	0.0947	0.0830	0.0651	0.0558
$ \mathbf{L} $	0.0647	0.0586	0.0401	0.0253
$\mathbf{H}_t$	0.0582	0.0494	0.0299	0.0175
$\mathbf{D}_k$	0.0607	0.0523	0.0385	0.0225

Table 4

Relative deviation of spectral distances for random graphs

that we can carefully control the edit operations. We generate 50 different graphs, and each graph represents a different class. We then create graphs to be classified by performing random edit operations on the class graphs. The graphs are classified using a simple 1-NN classifier and the Euclidean distance between the spectra; the aim here is to investigate the efficacy of the representation rather than the classifier. Figure 5 shows the classification error rates over a range of numbers of edit operations. These results clearly show the superiority of the Laplacian. The adjacency matrix is a poor representation. The normalised Laplacian is significantly worse than the Laplacian and gives the same performance as the heat kernel, whereas the path-length distribution is worse.

## 6.2 Shape representation

In order to furnish a set of experiments on trees, we have used the shape database of Torsello and Hancock[21]. Some examples are shown in Figure 6. These shapes are skeletonised using the medial axis transform. The resulting skeleton forms a tree with vertices at the junctions. The edge weights are determined by the arc length of the skeleton. We perform two experiments on this data; the first is a clustering experiment. We use the feature space of the graph eigenvalues and compute a cluster compactness index for shapes from

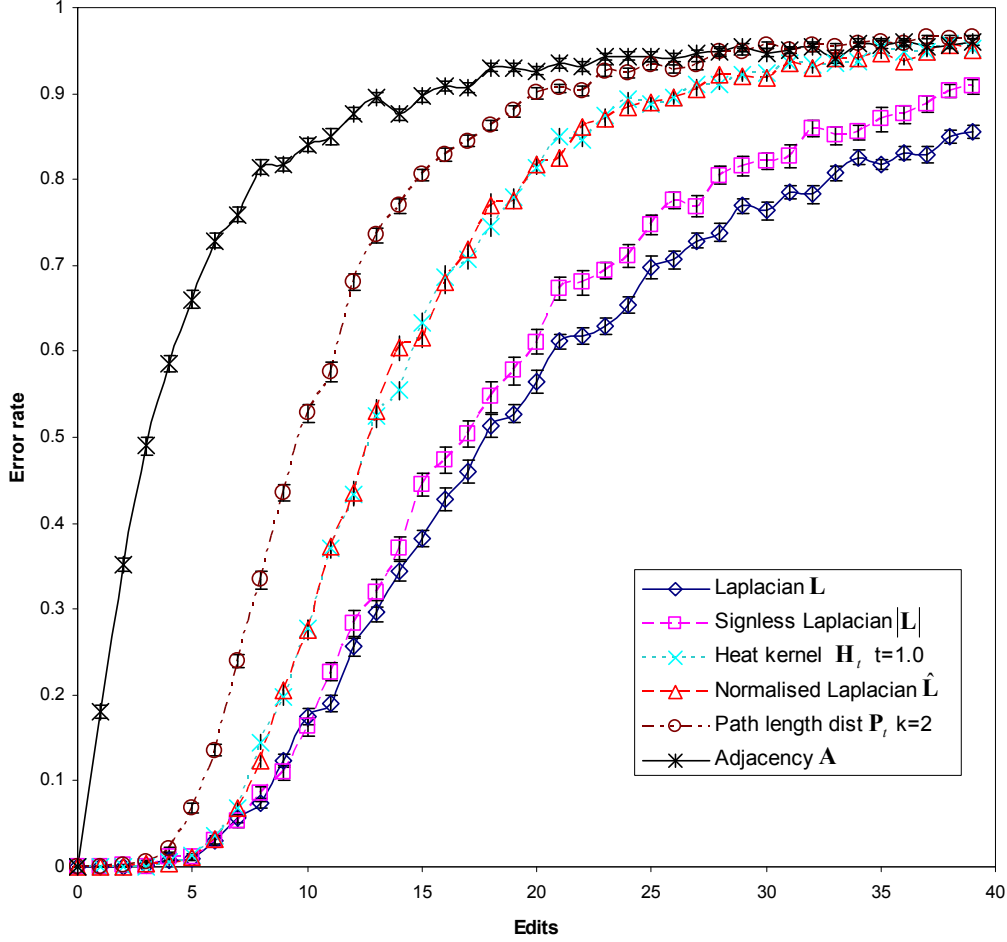


Fig. 5. Error rate for spectral graph classification for six methods of matrix representation and random graphs

three different classes in the dataset. We use three shapes from the database in order to get a clustering problem which is of moderate difficult for the algorithm. A problem which is too easy (too few clusters) or too hard (too many clusters) will not produce a result which is sensitive to changes in representation. Ideally, the representation should group shapes from the same class together, giving compact clusters. Cluster quality is measured by an index related to the Davies-Bouldin measure:

$$\mathcal{M} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \max_{j \in \mathcal{C}, j \neq i} \frac{\frac{1}{n_i^2} \sum_{k, l \in C_i} d(\mathbf{x}_k, \mathbf{x}_l) + \frac{1}{n_j^2} \sum_{k, l \in C_j} d(\mathbf{x}_k, \mathbf{x}_l)}{\frac{2}{n_i n_j} \sum_{k \in C_i, l \in C_j} d(\mathbf{x}_k, \mathbf{x}_l)} \quad (8)$$

Here  $x_k$  is the vector of eigenvalues representing point  $k$ , and  $d(\cdot, \cdot)$  is the Euclidean distance.  $C_i$  is the set of points belonging to cluster  $i$ ,  $n_i$  is the number of points in that cluster, and  $\mathcal{C}$  is the set of clusters. This measure gives zero for a perfect cluster separation and one for complete overlap of the



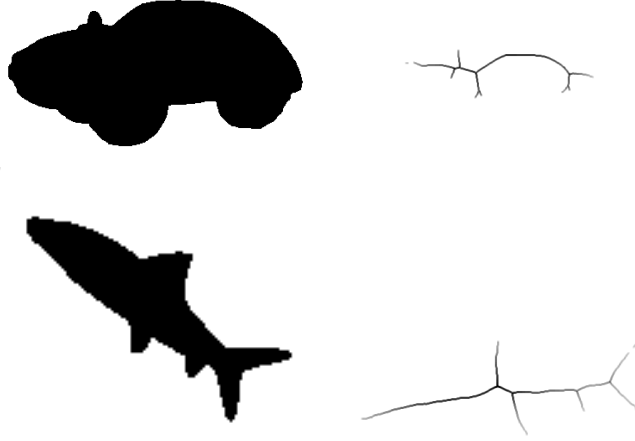


Fig. 6. Example shapes and their skeletons from the shape database.

Matrix	Class. error rate	Cluster compactness
<b>L</b>	0.56463	0.56615
<b>A</b>	0.63265	0.56487
$\hat{\mathbf{L}}$	0.67346	0.49028
$\mathbf{H}_t(t = 0.1)$	0.67346	0.49026
$\mathbf{D}_k(k = 4)$	0.69387	0.52642

Table 5

Clustering and classification performance of graph spectra on shape trees

clusters, and is based on the ratio of within-cluster distances to intra-cluster distances.

We also attempt to classify the shapes using a 3NN classifier on the graph spectrum. Here we use the full set of 10 shape classes, with 15 examples in each class. The results are shown in Table 5

Again the Laplacian performs well, but the picture here is much more mixed. The adjacency matrix does much better, and while the other representations give worse classification, they give tighter clusters.

### 6.3 Image graphs

Our final set of experiments employs images from the COIL database[13]. We use four objects which have 72 views each from varying directions (Objects 1-4, Figure 7. A corner detector is applied to the images and the resulting points are converted into a Delaunay graph[28]. Again we apply the same clustering measure and classifier to this dataset as we used for the shapes. The results

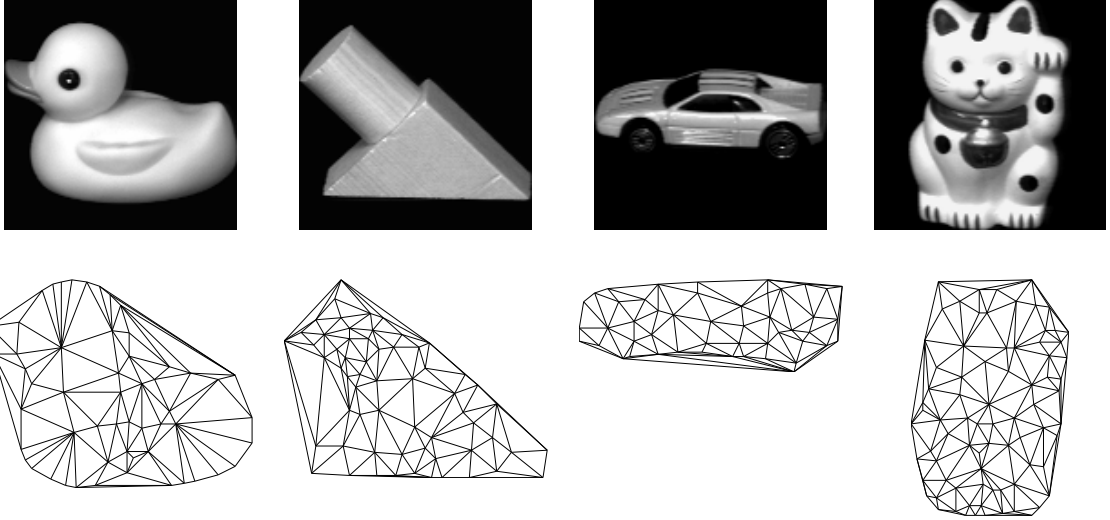


Fig. 7. Example images of the objects used in the COIL database, and their Delaunay graphs

Matrix	Class. error rate	Cluster compactness
$\mathbf{L}$	0.35763	0.92577
$ \mathbf{L} $	0.39931	0.94282
$\mathbf{D}_k(k=2)$	0.44097	0.94407
$\hat{\mathbf{L}}$	0.46528	0.94499
$\mathbf{H}_t(t=0.1)$	0.46528	0.94532
$\mathbf{A}$	0.63542	0.99614

Table 6  
Clustering and classification performance of graph spectra on the COIL database

are shown in Table 6.

For this dataset, **the Laplacian is superior in all regards.** The pattern here is much more similar to that shown in the synthetic dataset; the adjacency matrix performs poorly, with the normalised Laplacian and derived representations performing between the two extremes.

In summary, for classification and clustering problems, our results show that the Laplacian representation is the best one to use. It produces lower error rates on classification problems with synthetically generated graphs, shock graphs and Delaunay graphs. The results for clustering are more ambiguous, with the Laplacian best for Delaunay graphs and the heat kernel best for shock graphs. This is in contrast to the results for cospectrality and isomorphism, that show that the normalised Laplacian reduced problems of cospectrality. The normalised Laplacian seems to introduce more variability in the spectra, but this variability does not reflect structural similarity.

## 7 Conclusions

Our results show that use of the Laplacian matrix or its derivatives can drastically reduce the problem of cospectrality between trees. If the trend we have seen continues, then virtually all trees of a small size will have a unique Laplacian spectrum. On the other hand, it is known that nearly all trees are cospectral, when the trees are large enough. It is clear from our results that ‘large’ must be considerably larger than 25 vertices, and for trees of this size cospectrality is not likely to be a problem if a matrix from the Laplacian family is used.

For all these representations, there is a close relationship between the chosen edit distance and spectral distance between graphs. This relationship is most rigidly observed by the heat kernel.

As a graph similarity measure, the Laplacian matrix again seems superior. The adjacency matrix is the weakest representation of the ones investigated and its spectrum should not be used to represent the underlying graph. While the performance of the signless Laplacian is similar to that of the Laplacian itself, the normalised Laplacian and the related heat kernel and path-length distribution are notably less effective. This weighs against the use of the heat kernel, where an adjustable parameter is required without significant benefit.

## References

- [1] X. Bai and E. R. Hancock. Heat kernels, manifolds and graph embedding. In *IAPR workshop on structural, syntactic and statistical pattern recognition, LNCS 3138*, pages 198–206, 2004.
- [2] H. G. Barrow and R. M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4:83–84, 1976.
- [3] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18:689–694, 1997.
- [4] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922, 1999.
- [5] F. R. K. Chung. *Spectral Graph Theory*, volume CBMS series 92. American Mathematical Society Ed., 1997.
- [6] M. F. Demirci, R. van Leuken, and R. Veltkamp. Indexing through laplacian spectra. *Computer Vision and Image Understanding*, 2008.

- [7] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computing*, 22(1):67–92, 1973.
- [8] C. D. Godsil and B. D. McKay. Constructing cospectral graphs. *Aequationes Mathematicae*, 25:257–268, 1982.
- [9] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:377–388, 1996.
- [10] W. H. Haemers and E. Spence. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211, 2004.
- [11] S. Kosinov and T. Caelli. Inexact multisubgraph matching using graph eigenspace and clustering models. *Structural, Syntactic and Statistical Pattern Recognition, LNCS*, 2396:133–142, 2002.
- [12] G. Li and F. Ruskey. The advantages of forward thinking in generating rooted and free trees. In *10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 939–940, 1999.
- [13] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, 1996.
- [14] A. Robles-Kelly and E. R. Hancock. A riemannian approach to graph embedding. *Pattern Recognition*, 40:1042–1056, 2007.
- [15] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern-recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13(3):353–362, 1983.
- [16] S. Sarkar and K. L. Boyer. Preceptual organization in computer vision. *IEEE Transactions on Systems, Man and Cybernetics*, 23:382–399, 1993.
- [17] A. J. Schwenk. Almost all trees are cospectral. In F. Harary, editor, *New directions in the theory of graphs*, pages 275–307. Academic Press, New York, 1973.
- [18] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997.
- [19] A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral coding of topological structure. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, number 491–497, 1999.
- [20] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.
- [21] A. Torsello and E. R. Hancock. Learning shape-classes using a mixture of tree-unions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):954–967, 2006.

- [22] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- [23] E. R. van Dam and W. H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 356:241–272, 2003.
- [24] Barend J. van Wyk and Michaël A. van Wyk. A pocs-based graph matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1526–1530, 2004.
- [25] R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648, 1997.
- [26] R. C. Wilson and E. R. Hancock. Pattern spaces from graph polynomials. In *12<sup>th</sup> International Conference on Image Analysis and Processing*, pages 480–485, 2003.
- [27] R. C. Wilson and E. R. Hancock. Contour segments from spline interpolation. In *Syntactic and Structural Pattern Recognition Workshop*, volume LNCS 3138, pages 57–65. Springer, 2004.
- [28] B. Xiao and E. R. Hancock. Clustering shapes using heat content invariants. In *IEEE International Conference on Image Processing*, 2005.
- [29] S. T. Yau and R. M. Schoen. *Differential Geometry*. Science Publication, 1988.
- [30] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh, and S. J. Dickinson. Retrieving articulated 3-d models using medial surfaces and their graph spectra. In *EMMCVPR*, 2005.