



Coined quantum walks lift the cospectrality of graphs and trees

David Emms^a, Simone Severini^b, Richard C. Wilson^a, Edwin R. Hancock^{a,*}

^aDepartment of Computer Science, University of York, York YO10 5DD, UK

^bInstitute for Quantum Computing, Department of Combinatorics and Optimization, University of Waterloo, 2133 Davis Centre, Waterloo N2L 3G1, Ontario, Canada

ARTICLE INFO

Article history:

Received 9 June 2008

Received in revised form 6 October 2008

Accepted 15 October 2008

Keywords:

Graph spectra

Quantum walks

Unitary representations

Strongly regular graphs

Graph matching

Cospectrality

ABSTRACT

In this paper we explore how a spectral technique suggested by coined quantum walks can be used to distinguish between graphs that are cospectral with respect to standard matrix representations. The algorithm runs in polynomial time and, moreover, can distinguish many graphs for which there is no subexponential time algorithm that is proven to be able to distinguish between them. In the paper, we give a description of the coined quantum walk from the field of quantum computing. The evolution of the walk is governed by a unitary matrix. We show how the spectrum of this matrix is related to the spectrum of the transition matrix of the classical random walk. However, despite this relationship the behaviour of the quantum walk is vastly different from the classical walk. This leads us to define a new matrix based on the amplitudes of paths of the walk whose spectrum we use to characterise graphs. We carry out three sets of experiments using this matrix representation. Firstly, we test the ability of the spectrum to distinguish between sets of graphs that are cospectral with respect to standard matrix representation. These include strongly regular graphs, and incidence graphs of balanced incomplete block designs (BIBDs). Secondly, we test our method on ALL regular graphs on up to 14 vertices and ALL trees on up to 24 vertices. This demonstrates that the problem of cospectrality is often encountered with conventional algorithms and tests the ability of our method to resolve this problem. Thirdly, we use distances obtained from the spectra of $S^+(U^3)$ to cluster graphs derived from real-world image data and these are qualitatively better than those obtained with the spectra of the adjacency matrix. Thus, we provide a spectral representation of graphs that can be used in place of standard spectral representations, far less prone to the problems of cospectrality.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Quantum algorithms have recently attracted considerable attention in the theoretical computer science community. This is primarily because they offer a considerable speed-up over classical algorithms. For instance, Grover's [1] search method is polynomially faster than its classical counterpart, and Shor's factorisation method [2] is exponentially faster than known classical methods. However, quantum algorithms also have a richer structure than their classical counterparts, since they use qubits rather than bits as the basic representational unit [3]. For instance, this structure is exploited in Shor's algorithm where the Fourier transform is used to locate prime factors. The interference and entanglement of qubits may also be exploited to develop interesting protocols, and one fascinating example is Braunstein's quantum teleportation idea [4]. It is this issue of richer representations that is the subject of this paper. We are interested in the problem of random walks on graphs

and how they can be used to distinguish graphs which cannot be distinguished efficiently by conventional methods. Random walks are useful tools in the analysis of the structure of graphs [5]. The steady state random walk on a graph is given by the leading eigenvector of the transition probability matrix, and this in turn is related to the eigenstructure of the graph Laplacian [6]. Hence, the study of random walks has been the focus of sustained research activity in spectral graph theory. For instance, Lovász has written a useful review of the subject [6], and spectral bounds have been placed on the properties of random walks, including the mixing times and hitting times [7].

From a practical perspective, there have been a number of useful applications of random walks. One of the most important of these is the analysis of routing problems in network and circuit theory [8]. Of more recent interest is the use of ideas from random walks to define the page-rank index for Internet search engines such as Googlebot [9]. In the pattern recognition community there have been several attempts to use random walks for graph matching. These include the work of Robles-Kelly and Hancock [5,10] which has used both a standard spectral method [10] and a more sophisticated one based on ideas from graph seriation [5] to convert graphs to strings, so that string matching methods may be used to compare graphs. Sarti et al.

* Corresponding author. Tel.: +44 1904 43 3374; fax: +44 1904 43 2767.
E-mail address: erh@cs.york.ac.uk (E.R. Hancock).

[11], on the other hand, have used ideas borrowed from page-rank to associate a spectral index with graph nodes and have then used standard subgraph isomorphism methods for matching the resulting attributed graphs.

One of the problems that limits the use of random walks, and indeed any spectral method, is that of cospectrality. This is the situation in which structurally distinct graphs present the same pattern of eigenvalues. Classic examples are strongly regular graphs (SRGs) [12] and almost all trees [13,14]. The cospectrality of coparametric SRGs is due to the high level of symmetry that they share. The parameters describe the regularity of the graph and the number of vertices that any pair of vertices will both be adjacent to. In addition, almost all trees have a cospectral partner. In these cases conventional methods will fail to distinguish between such non-isomorphic graphs.

Recently, quantum walks have been introduced as quantum counterparts of random walks [15,16]. Their behaviour is governed by unitary rather than stochastic matrices. The stochastic matrix of a classical random walk is such that its columns sum to unity. A unitary matrix, on the other hand, has complex entries. For a unitary matrix the squares of the entries in the columns sum to unity. Quantum walks possess a number of interesting properties not exhibited by classical random walks. For instance, because the evolution of the quantum walk is unitary and therefore reversible, the walks are non-ergodic, and what is more, they do not have a limiting distribution. Practical applications of quantum walks are fast quantum algorithms for database searching [17], graph traversal [18,19], and the problem of element distinctness [20]. Although the analysis of quantum walks may seem detached from the practical problems listed above, they may offer a way of countering the problem of cospectrality as we will show later in this paper.

The main contribution of this paper is to use the unitary matrix, U , governing the coined quantum walk to construct a new representation for graphs that is able to overcome the problem of cospectrality in many instances where it is encountered. To construct U we make use of ‘Grover coins’ to determine the quantum amplitudes for the transitions between adjacent states. The (i,j) th entry of U^3 represents the sum of the quantum amplitudes of the paths of length 3 from the state j to the state i of the walk. Since the amplitudes can be positive as well as negative, this sum depends crucially on the destructive and constructive interference of the various paths. The positive support of the matrix U^3 is the 0–1 matrix with an (i,j) th entry of 1 if and only if the (i,j) th entry of U^3 is positive. We use the ordered eigenvalues of the positive support of U^3 as a graph invariant to represent the graph. We show that this representation overcomes the problem of cospectrality in many instances in which it normally occurs. We also show that the representation is robust enough to replace the use of the adjacency matrix in the case where spectral methods are used for inexact graph matching. In addition, the work demonstrates how classical algorithms can utilise the often richer quantum representations in order to provide new ways of approaching common problems.

The remainder of the paper is organised as follows. In Section 2.2 we provide a brief overview of quantum computing. In Section 2.3 we describe the classical random walk, the quantum random walk and contrast the two. In Section 3 we examine the spectrum of the matrix governing the quantum random walk and show that it is not a solution to the problem of cospectrality. This leads us to consider interference effects and thus to propose and develop a method based on the positive support of the cubed unitary matrix. In Section 3.2 we examine the effectiveness of this approach using SRGs and thereby define our matrix representation. In Section 4 we detail the experiments we have carried out using this matrix representation and in Section 5 we present the conclusions that we draw from our work.

2. Background material

In this section we present some necessary background material for the work presented. We begin by introducing some basic graph theory. We then provide an introduction to quantum computing. Finally, we introduce the classical random walk and discrete-time quantum walk.

2.1. Graphs

A graph is a pair, $G = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of vertices and \mathcal{E} a set of unordered pairs, $\{u, v\}$, such that $u, v \in \mathcal{V}$. An element of \mathcal{E} is called an edge and if $\{u, v\} \in \mathcal{E}$ we say that u and v are adjacent and write this as $u \sim v$. Let $n = |\mathcal{V}|$ be the number of vertices in the graph. The neighbourhood, $\mathcal{N}(u) = \{v | u \sim v\}$, of a vertex u is the set of vertices that are adjacent to it. A weighted graph is a graph together with a weight function, $W : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$, such that $W(u, v) = W(v, u)$, and $W(u, v) \neq 0$ if and only if u is adjacent to v . Of the two, a weighted graph is more general since any unweighted graph can be a weighted graph by defining the weight function to be such that $W(u, v) = 1$ if $u \sim v$ and $W(u, v) = 0$ otherwise. Thus, the definitions that we give below will be for weighted graphs.

The structure of a graph is often represented using its adjacency matrix, A , whose entries are given by

$$A_{uv} = \begin{cases} W(u, v) & \text{if } \{u, v\} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

The degree of a vertex $u \in \mathcal{V}$, denoted $d(u)$, is given by

$$d(u) = \sum_{v: v \sim u} W(u, v).$$

For an unweighted graphs, the degree of a vertex is simply the number of vertices adjacent to it. The matrix $D = \text{diag}(d(1), d(2), \dots, d(n))$, with the vertex degrees on the diagonal and zeros elsewhere, is referred to as the degree matrix. We say that an unweighted graph is regular if the degree of every vertex is the same. The Laplacian matrix of a graph, $L = A - D$, can be calculated from the degree and adjacency matrices and has entries

$$L_{uv} = \begin{cases} W(u, v) & \text{if } \{u, v\} \in \mathcal{E}, \\ -d(u) & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

To write down any of these matrices for a graph requires an implicit numbering of the vertices. However, there is no method for uniquely numbering the vertices of a graph; a particular labelling of the vertices is not a property of a graph. Thus, an isomorphism between two graphs, $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$, is a one-to-one mapping between the vertex sets, $\zeta : \mathcal{V}_G \rightarrow \mathcal{V}_H$, such that $g_1 \sim g_2$ if and only if $\zeta(g_1) \sim \zeta(g_2)$.

2.2. Quantum computing

The field of quantum computing is concerned with the manipulation of quantum states in order to carry out computation. Whereas a classical computer manipulates classical states, a quantum computer would manipulate quantum states, such as the polarisation of photons [21] or the excitation states of trapped ions [22].

2.2.1. Mathematical description of quantum computing

A state of a quantum mechanical system is described by a vector of unit length in a complex-valued Hilbert space, \mathcal{H} . If the Hilbert space is n -dimensional then $\mathcal{H} \cong \mathbb{C}^n$. We use Dirac’s notation and write the basis for such a space as $\{|j\rangle | 1 \leq j \leq n\}$, where the $|j\rangle$ are

orthonormal unit vectors in \mathcal{H} . A general state of the system is a unit-length, complex-linear sum of basis states. Such a state is referred to as a *superposition* of the basis states. That is, a general state, $|\psi\rangle \in \mathcal{H}$, is of the form

$$|\psi\rangle = \sum_{j=1}^n a_j |j\rangle,$$

where $a_j \in \mathbb{C}$ is referred to as the *quantum amplitude* of the state $|j\rangle$.

The inner product is given by the standard inner product on \mathbb{C}^n . That is, given a state

$$|\psi_a\rangle = \sum_{j=1}^n a_j |j\rangle$$

in \mathcal{H} , we write as $\langle\psi_a|$ the linear functional that maps any vector,

$$|\psi_b\rangle = \sum_{j=1}^n b_j |j\rangle$$

to the standard inner product. Thus, the inner product, written as $\langle\psi_a|\psi_b\rangle$, is given by

$$\langle\psi_a|\psi_b\rangle = (\langle\psi_a|, |\psi_b\rangle) = \sum_{j=1}^n a_j^* b_j,$$

where a_j^* is the complex conjugate of a_j .

The probability of the state $|\psi\rangle$ being in a particular basis state $|j\rangle$ (i.e. the probability that j is the outcome of a measurement of the state) is given by

$$\Pr(j) = |\langle j|\psi\rangle|^2. \tag{1}$$

Thus the requirement that states are represented by vectors of unit length corresponds to the requirement that the sum of the probabilities of all possible outcomes is one. Note that two states are equivalent if they differ only by a phase factor. That is $|\psi\rangle$ and $e^{i\alpha}|\psi\rangle$, $\alpha \in \mathbb{R}$, represent indistinguishable states, since they give rise to the same probabilities for all possible outcomes.

Since probability must be conserved and since the evolution of closed quantum systems is linear (in the absence of any measurements), the time evolution of a quantum system is described by unitary operators. That is, if the system is in the state $|\psi_{t_1}\rangle$ at time t_1 and $|\psi_{t_2}\rangle$ at time t_2 , then $|\psi_{t_2}\rangle = U|\psi_{t_1}\rangle$ for some unitary operator U . Thus a quantum computation would consist of the application of some finite sequence of elementary unitary operators. The analogue of the bit in the field of quantum computation is the *qubit*. A qubit is a two-dimensional quantum state whose basis we write as $\{|0\rangle, |1\rangle\}$ and a general state is of the form

$$|\psi\rangle = a|0\rangle + b|1\rangle,$$

where $a, b \in \mathbb{C}$ and $aa^* + bb^* = 1$.

2.2.2. Some differences between quantum and classical computation

A state of an n bit classical computer at any given point in time is one of the 2^n possible strings of length n of zeros and ones. The state of an n qubit quantum computer is, on the other hand, a superposition of such basis states. That is, it is a complex linear combination of all the 2^n possible basis states (where up to $2^n - 1$ of the states can have zero amplitude). Thus, a quantum computation, or equivalently an operation on a quantum state with a unitary operator, can be used to calculate the result of the computation on all of the 2^n basis states simultaneously. However, not all of the information contained in the quantum state is accessible. When a measurement is

made only one of the possible basis states is observed. The probabilities for observing each of the possible states are given by Eq. (1). If the state $|j\rangle$ is observed then the post-measurement state is $|\psi\rangle = |j\rangle$. That is, all further information previously contained in the state is lost and further measurements will not be able to extract any more information about the pre-measurement state.

Algorithms have been designed that take advantage of the mechanics of quantum systems to solve problems quadratically [1] or even exponentially quicker [2] than the best classical algorithms. However, designing algorithms that effectively utilise the often counter-intuitive properties of quantum mechanics has proven difficult. In order to facilitate their design, the quantum walk (the quantum analogue of the random walk) has been developed. In what follows we give an overview of both the classical random walk and the discrete-time quantum walk.

2.3. Random walks

A random walk consists of two components: a state space on which at any point in time there is a probability distribution giving the location of the walk and a transition function which gives the probabilities for transitions between one state and another. As such, random walks are most naturally defined on graphs since the connectivity structure of a graph defines which transitions are allowed. Additionally, weighted graphs can be used if the walk is to be biased in some way. In what follows we present the classical random walk and the discrete-time quantum walk, and then elucidate some of the differences between the two.

2.3.1. Classical random walks

Random walks are a model of diffusion which are important in, amongst other areas, statistical physics, applied probability and randomised algorithms [6,7,23]. A good survey is provided by Lovász [6].

The state space for the (discrete-time) classical random walk on a graph, $G = (\mathcal{V}, \mathcal{E})$, is the set of vertices, \mathcal{V} . The walk at vertex u at time t moves to an adjacent vertex, v , at time $t + 1$ with probability $W(u, v)/d(u)$ and to any non-adjacent vertex with zero probability. The sequence of vertices visited is a finite Markov chain. The state space for the walk is thus \mathbb{R}^n , and the basis state $\mathbf{e}_u = (0, \dots, 0, 1, 0, \dots, 0)^T$, i.e. the vector with a one as its u th entry and zeros elsewhere, corresponds to the walk being at vertex u with probability one. A general state for the walk at time t is described by a probability distribution vector, $\mathbf{p}_t \in \mathbb{R}^n$, whose u th entry gives the probability that the walk is at vertex u at time t . The *transition matrix* for the walk is a stochastic matrix, that is, its entries are from the interval $[0, 1]$ and its columns sum to one. It is given by $T = AD^{-1}$ and has entries

$$T_{uv} = \begin{cases} \frac{W(u, v)}{d(v)} & \text{if } \{u, v\} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

The evolution of the probability distribution vector is given by $\mathbf{p}_{t+1} = T\mathbf{p}_t$, and thus we have that $\mathbf{p}_t = T^t\mathbf{p}_0$. Therefore, the probability that the random walk currently at vertex u will be at vertex v after t steps is given by $(T^t)_{vu}$. As \mathbf{p}_t is a probability distribution we require that $\sum_{u \in \mathcal{V}} p_t(u) = 1$ for all times $t \in \mathbb{R}^+$. Note that T is defined such that if this is true at $t = 0$ then it will be true for all $t \in \mathbb{Z}$.

The state vector $\boldsymbol{\kappa}$, whose u th entry is given by

$$\boldsymbol{\kappa}(u) = \frac{d(u)}{2|\mathcal{E}|}$$

is a stationary state of the walk. That is to say that, $\boldsymbol{\kappa} = T\boldsymbol{\kappa}$. Provided that the graph is connected and is not bipartite, then regardless of the starting state, $\mathbf{p}_t \rightarrow \boldsymbol{\kappa}$ as $t \rightarrow \infty$. If G is a regular graph then the

random walk is symmetric in the sense that $T_{ij} = T_{ji}$. Basic properties of the random walk such as the mean hitting time (the mean time for the walk to first hit a particular vertex) and the mixing rate (the rate at which the walk tends towards the limiting distribution) are determined by the spectrum of T . A review of these spectral properties is given in Ref. [6].

2.3.2. The discrete-time quantum walk

The discrete-time quantum walk is the quantum counterpart of the discrete-time classical random walk and has been used in the design of new quantum algorithms [15,16,24,25]. Quantum processes are reversible, and in order to make the discrete-time quantum walk reversible a particular state must specify both the current and previous location of the walk [24]. To this end each edge of the graph, $\{u, v\} \in \mathcal{E}$, is replaced by a pair of (directed) arcs, (u, v) and (v, u) , and the set of these arcs is denoted by \mathcal{A} . The state space for the discrete-time quantum walk is the set of arcs, \mathcal{A} . If the walk is at vertex v having previously been at vertex u with probability 1, then the state is written as $|\psi\rangle = |uv\rangle$. Transitions are possible from one arc (w, x) to another arc (u, v) (i.e. from a state $|wx\rangle$ to $|uv\rangle$) if and only if $x = u$ and $x \sim v$. Note that this corresponds to only permitting transitions between adjacent vertices. The state vector for the walk is a quantum superposition of states on single arcs of the graph, and can be written as

$$|\psi\rangle = \sum_{(u,v) \in \mathcal{A}} \alpha_{uv} |uv\rangle,$$

where the quantum amplitudes are complex, i.e. $\alpha_{uv} \in \mathbb{C}$. Using Eq. (1), the probability that the walk is in the state $|uv\rangle$ is given by $\text{Pr}(|uv\rangle) = \alpha_{uv} \alpha_{uv}^*$.

As with the classical walk, the evolution of the state vector is determined by a matrix, in this case denoted U , according to $|\psi_{t+1}\rangle = U|\psi_t\rangle$. Since the evolution of the walk is linear and conserves probability the matrix U must be unitary. That is, the inverse is equal to the complex conjugate of the matrix transposed, i.e. $U^{-1} = U^\dagger$. The entries of U determine the probabilities for transitions between states. Thus, there are constraints on these entries and there are therefore constraints on the permissible amplitudes for the transitions. The sum of the squares of the amplitudes for all the transitions from a particular state must be unity. Consider a state $|\psi\rangle = |u_1 v\rangle$ where the neighbourhood of v , $\mathcal{N}(v) = \{u_1, u_2, \dots, u_r\}$. A single step of the walk should only assign non-zero quantum amplitudes to transitions between adjacent states, i.e. the states $|vu_i\rangle$ where $u_i \in \mathcal{N}$. However, since U must be unitary these amplitudes cannot all be the same. Recall that the walk does not rely on any labelling of the edges or vertices. Thus, the most general form of transition will be one that assigns the same amplitudes to all transitions $|u_1 v\rangle \rightarrow |vu_i\rangle$, $u_i \in \mathcal{N} \setminus u_1$, and a different amplitude to the transition $|u_1 v\rangle \rightarrow |vu_1\rangle$. The second of these two transitions corresponds to the walk returning along the same edge to which it came. Thus, the transition will be of the form

$$|u_1 v\rangle \rightarrow a|vu_1\rangle + b \sum_{i=2}^r |vu_i\rangle, \quad a, b \in \mathbb{C}.$$

It is usual to use the Grover diffusion matrices [1] which assign a quantum amplitudes of $a = 2/d(v) - 1$ when the walk returns along the same edge and $b = 2/d(v)$ for all other transitions. Such matrices are used as they are the matrices furthest from the identity which are unitary and are not dependent on any labelling of the vertices.

Using the Grover diffusion matrices, the matrix, U , that governs the evolution of the walk has entries

$$U_{(u,v),(w,x)} = \begin{cases} \frac{2}{d(x)} - \delta_{vw} & \text{if } u = x, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

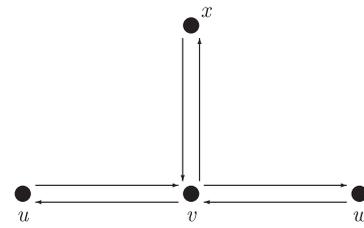


Fig. 1. A simple graph that demonstrates one of the effects of quantum interference on the discrete-time quantum walk.

for all $(u, v), (w, x) \in \mathcal{A}$, where δ_{vw} is the Kronecker delta. The $(u, v), (w, x)$ entry of this matrix gives the quantum amplitude for transition $(w, x) \rightarrow (u, v)$. That is, the transition from the vertex x (having previously been at w) to the vertex v (having previously been at u). Note that although the entries are real, they are negative as well as positive. Consequently, the quantum amplitude for a state can be negative. This is of key importance as it allows *destructive interference* to take place. To see how destructive interference can affect the discrete-time quantum walk, consider the graph in Fig. 1. Destructive interference can occur when, say, we have the state

$$|\psi_t\rangle = a|uv\rangle - b|vw\rangle.$$

After one step of the walk the state will be

$$|\psi_{t+1}\rangle = \begin{cases} -\frac{1}{3}|vu\rangle + \frac{2}{3}|vw\rangle + \frac{2}{3}|vx\rangle & \text{if } a = 1 \text{ and } b = 0, \\ \frac{2}{3}|vu\rangle - \frac{1}{3}|vw\rangle + \frac{2}{3}|vx\rangle & \text{if } a = 0 \text{ and } b = 1, \\ -\frac{1}{\sqrt{2}}|vu\rangle + \frac{1}{\sqrt{2}}|vw\rangle + 0|vx\rangle & \text{if } a = b = 1/\sqrt{2}. \end{cases}$$

Considering the amplitude of the state $|vx\rangle$ at time $t + 1$. If $a = 1$ and $b = 0$ then the walk is in the state $|uv\rangle$ with non-zero probability initially (with probability 1 in fact) and at state $|vx\rangle$ with non-zero probability at time $t + 1$. If $a = 0$ and $b = 1$ then the walk is in the state $|vw\rangle$ with non-zero probability initially and at state $|vx\rangle$ with non-zero probability at time $t + 1$. However, if $a = b = 1/\sqrt{2}$ then the walk is at either $|uv\rangle$ or $|vw\rangle$ both with non-zero probability but at $t + 1$ the probability of being at $|vx\rangle$ is zero. That is, since the evolution of the walk is linear in terms of the amplitudes and not the probabilities, these amplitudes can cancel out. The result is that two paths, which individually would both lead to a non-zero probability of the walk being observed at $|vx\rangle$, can cancel out, and the walk is not observed at $|vx\rangle$. The classical walk, where the transitions are linear with respect to the (always positive) probabilities, does not possess this property.

2.3.3. The classical and quantum walks contrasted

The behaviour of the classical and quantum walks differs significantly. This section attempts to highlight some of those differences by considering the walk on the line. For the quantum walk on the line [26], the basis states can be written as $|nd\rangle$, where $n \in \mathbb{Z}$ indicates the present position and $d = l, r$ indicates whether the walk reached that position by approaching it from the left, l , or the right, r (clearly this is equivalent to specifying the vertex from which the walk arrived). The particular case of the walk on the line using the amplitudes given by the Grover diffusion matrices does not produce an interesting walk. The reason for this is that since the degree of every vertex is two, the walk will move in just one direction (see Eq. (2)). To best compare the quantum walk with its classical counterpart we use the symmetric starting state $|\psi_0\rangle = 1/\sqrt{2}(|0l\rangle + |0r\rangle)$ and the Hadamard coin. The Hadamard coin is such that applied to

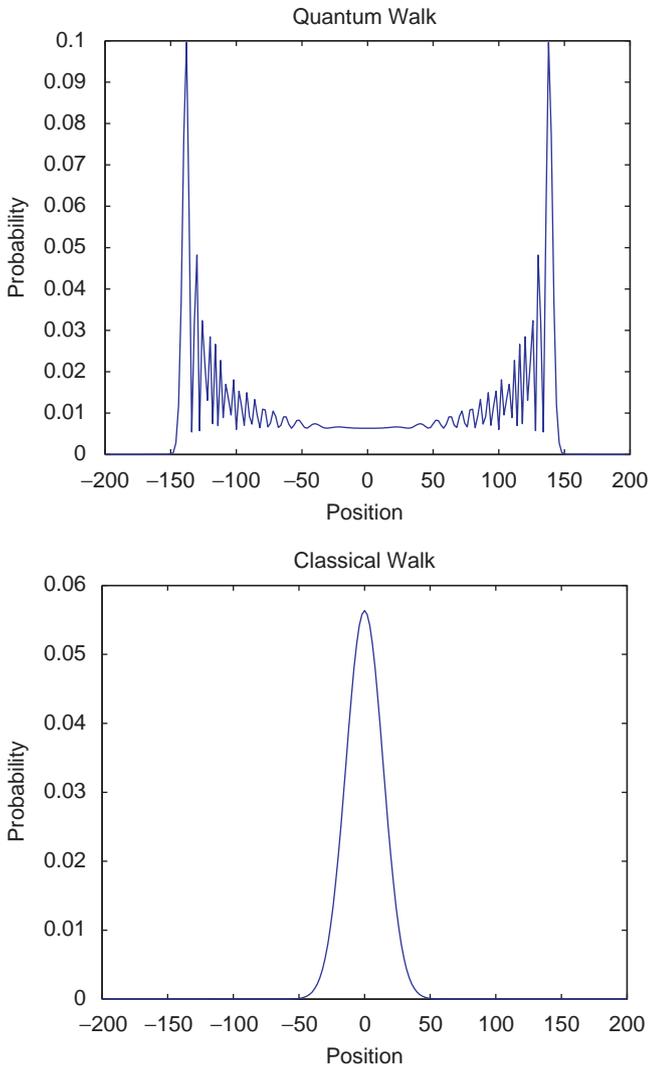


Fig. 2. The probability distribution for the quantum walk on the line using the Hadamard coin and a symmetric starting state (top) and the classical walk (bottom). Only odd positions are plotted since even positions have probability zero.

the states $|nr\rangle$ and $|nl\rangle$ ($n \in \mathbb{Z}$) we have

$$|nr\rangle \rightarrow \frac{1}{\sqrt{2}}(|(n-1)r\rangle + |(n+1)l\rangle)$$

and

$$|nl\rangle \rightarrow \frac{1}{\sqrt{2}}(|(n-1)r\rangle - |(n+1)l\rangle).$$

If the walk is allowed to evolve without being measured then after 200 steps the probability distribution is as shown in Fig. 2. The distribution is in stark contrast to that observed for the classical walk (also shown in Fig. 2), which tends towards a Gaussian distribution with mean $\mu=0$ and variance $\sigma^2=n$, where n is the number of steps. The quantum walk has mean $\mu=0$ and variance $\sigma^2 \sim n^2$. It is almost uniform in the interval $[-n/\sqrt{2}, n/\sqrt{2}]$ and heavily peaked, with the modal positions close to the limits of the distribution.

An observation that can be made of the walk on the line is that the quantum walk spreads quadratically faster than the classical walk. It turns out that this is generally the case for walks on graphs [24]. However, there are cases where there is an exponential speed-up in the hitting time of a certain vertex in some graphs. This was observed on the hypercube by Kempe [19], and also for graphs constructed by joining two n -level binary trees at their leaves by Childs et al. [18].

3. Spectral analysis of U

We wish to concentrate on how the model of the discrete-time quantum walk can be used classically. Specifically, we are interested in being able to distinguish between graphs which are cospectral with respect to standard matrix representations. The spectrum of the unitary matrix governing the evolution of a discrete-time quantum walk turns out to be related to the spectrum of the transition matrix for the classical random walk. The unitary matrix governing the quantum walk, U , can be written in an expanded form, \tilde{U} , with entries

$$\tilde{U}_{(u,v),(w,x)} = A_{wx}A_{uv}\delta_{vw} \left(\frac{2}{d(x)} - \delta_{ux} \right)$$

for all $u, v, w, x \in \mathcal{V}$ (rather than all $(u, v), (w, x) \in \mathcal{A}$). The eigenvalues of \tilde{U} will be the same as the eigenvalues of U but with the addition of $|\mathcal{V}|^2 - 2|\mathcal{E}|$ eigenvalues of value 0 due to the additional zeros in the matrix \tilde{U} .

Let \mathbf{y} be an eigenvector of T with eigenvalue s , then \tilde{U} has a pair of eigenvalues $t^\pm = s \pm i\sqrt{1-s^2}$ with corresponding eigenvector \mathbf{z} , with entries

$$z_{(u,v)} = A_{uv} \left(\frac{y_w}{d(w)} - t^* \frac{y_x}{d(x)} \right). \tag{3}$$

A proof for this can be found in [27] our accompanying technical report [27]. This accounts for $2|\mathcal{V}|$ eigenvalues. The remaining $2|\mathcal{E}| - 2|\mathcal{V}|$ non-zero eigenvalues are ± 1 each with multiplicity $|\mathcal{E}| - |\mathcal{V}|$. This completes the spectrum of \tilde{U} and thus U . That is, the spectrum of U is completely determined by the spectrum of the transition matrix, T . The random walk induced by U has advantages over its classical analogue. In the next section we consider how to make use of the differences between the quantum walks on different graphs in order to distinguish between them.

3.1. Interference on the discrete-time quantum walk

The states of both the classical and the quantum walks are represented by state vectors. However, the paths of the classical walk can always be considered separately from each other. This is because the total probability is simply the sum of the probabilities for all the possible paths. This is not the case for the quantum walk. If there is a path between a pair of states with an associated non-zero quantum amplitude, it does not necessarily mean that there will be a non-zero probability of observing the walk at the terminal vertex of this path. An example of this was seen in the previous section. Examining the entries of the unitary matrix we see that negative amplitudes arise when a walk in the state $|uv\rangle$ moves to the state $|vu\rangle$ (provided that $d(v) > 2$). These negative amplitudes give rise to quantum interference which allows paths to interfere both constructively and destructively, and it is possible for two or more paths to exactly cancel each other out. The result is that even though there might exist a path of length t between a pair of states, in some cases there will be zero probability of such a transition since other paths may cancel it out. Therefore, the probability distribution on states of the walk after a given number of steps will depend not only on the number of paths between those states, but also on the interference between these paths. This allows the walk to probe and distinguish between graphs more effectively than is the case classically as we will demonstrate later in this section.

So far we have referred to the rows and columns of U by ordered pairs of vertices $(u, v) \in \mathcal{A}$. However, we can equally well refer to them with a single label, $j=1, 2, \dots, |\mathcal{A}|$, by numbering the arcs of the graph. The state of the quantum walk after t steps is given by $U^t|\psi_0\rangle$. The (j, k) th entry of U^t gives the quantum amplitude associated with moving from the state (arc) k to the state (arc) j in t steps, where

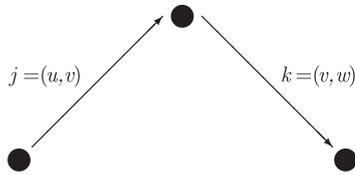


Fig. 3. The diagram shows two states on the quantum walk, the state j which represents the walk at vertex v , having previously been at u , and the state k for the walk at vertex w , having previously been at v . A transition is possible from the state j to the state k but not the other way around.

each of these states is an ordered pair of vertices in the graph. Let us consider the positive support [28], $S^+(V)$ of a matrix V , which is the matrix with entries

$$S^+(V)_{jk} = \begin{cases} 1 & \text{if } V_{jk} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

We begin by considering the positive support of powers of T , the transition matrix for the classical random walk. A non-zero (u, v) th entry indicates that there is a non-zero probability of the walk moving from vertex v to vertex u in t steps. For the classical walk this is the case if there is a path of length t between the vertices in the original graph. We can view $S^+(T^t)$ as an adjacency matrix of a graph, related to the original graph, with vertices representing the states of the walk (in this case simply vertices of the original graph). Thus, a pair of vertices is connected in this new graph if there is a path of length t between them.

The quantum walk, however, is different. The matrix $S^+(U^t)$ can be pictured as an adjacency matrix but in this case the adjacency matrix of a digraph and with vertex set \mathcal{A} . The graph is directed since, as shown in Fig. 3, a path of length t may be possible from a state j to a state k , but not necessarily from k to j . However, it is not sufficient for there simply to be a path of length t between state k and state j for there to be a non-zero (j, k) th entry in $S^+(U^t)$. The matrix has a non-zero (j, k) th entry if and only if there is a positive quantum amplitude for the walk starting at the state k being observed in the state j after t steps. A path of length t between the states is necessary, but it is not sufficient for this to occur. If there is a positive quantum amplitude then, in this digraph picture, there will be an arc from state j to state k .

The probability of a transition between two states is given by the square of the associated quantum amplitude. Thus, if there is a negative quantum amplitude associated with a particular transition then there is a non-zero probability that such a transition is possible. What is more, certain transitions which are possible for the classical random walk can in some cases have zero probability in the quantum walk due to destructive interference causing different paths between the two states to cancel out exactly. The regularity of some sets of graphs is such that, when we examine the eigenvalues of the matrix representing possible paths in the quantum walk, $S^+(T^t)$, they will be the same for all the graphs. Using $S^+(U^t)$ on the other hand, we chose to concentrate on a particular set of transitions, those that have a positive quantum amplitude. By focusing on this particular subset of the possible transitions, we draw more strongly on the connectivity structure of the graph in question. By doing so, despite the regularity of some graphs, we are able to distinguish between them in cases where conventional spectral methods fail to do so.

3.2. Strongly regular graphs

As mentioned earlier, SRGs can cause problems when we attempt to characterise graphs using their adjacency spectra due to their cospectrality. An SRG with parameters (n, k, λ, μ) is a k -regular

graph on n vertices such that every pair of adjacent vertices share λ common neighbours and every pair of non-adjacent vertices share μ common neighbours [12]. The spectra of the adjacency and Laplacian matrices of an SRG are completely determined by the SRGs parameters. However, large sets of coparametric non-isomorphic SRGs exist. In fact, the problem of deciding whether a pair of SRGs is isomorphic is a hard problem and the best known algorithm for testing whether SRGs are isomorphic runs in time $n^{O(n^{1/3})}$. This makes SRGs an ideal testing ground for any matrix representation that attempts to lift the cospectrality exhibited by some graphs. What is more, any method that is able to distinguish non-isomorphic SRGs in polynomial time would be important in its own right. Below we consider the spectra of $S^+(U^t)$ for small values of t and how this relates to the cospectrality problem.

3.2.1. Spectra of $S^+(U^t)$ for SRG

Let G be an SRG with parameters (n, k, λ, μ) , adjacency matrix A and transition matrix for the discrete-time quantum walk U . Let $\gamma_1, \gamma_2, \dots, \gamma_n$ be the eigenvalues of A . Then, the first $2n$ eigenvalues of $S^+(U)$ are

$$\delta_a^\pm = \frac{\gamma_a}{2} \pm i\sqrt{k-1-\gamma_a^2/4}, \quad a = 1, 2, \dots, n. \tag{4}$$

(See Ref. [27] for a proof.) We also observe (although we have not been able to prove) that the remaining $n(k-2)$ eigenvalues of $S^+(U)$ are ± 1 . Furthermore, we observe that the first $2n$ eigenvalues of $S^+(U^2)$ are

$$\delta_a^\pm = \frac{\gamma_a^2}{2} + 2 - k \pm i\gamma_a\sqrt{k-1-\gamma_a^2/4}, \quad a = 1, 2, \dots, n. \tag{5}$$

(See Ref. [27] for a proof.) We also observe (although we have not been able to prove) that the remaining $n(k-2)$ eigenvalues of $S^+(U^2)$ take the value 2. However, the spectrum of $S^+(U^3)$ is not determined by the parameters of the SRG. As an example, consider the pair of SRGs with parameters $(16, 6, 2, 2)$ shown in Fig. 4. The spectra of their adjacency matrices is given by

$$\text{sp}(A_G) = \text{sp}(A_H) = \{-2\}^9, [2]^6, [6],$$

and that of their Laplacian matrices by

$$\text{sp}(L_G) = \text{sp}(L_H) = \{0\}^1, [4]^6, [8]^9,$$

where $[a]^b$ indicates that the eigenvalue a has multiplicity b in the spectrum. However,

$$\text{sp}(S^+(U_G^3)) = \{-7-2i\}^{15}, [-7+2i]^{15}, [-5]^9, [-1]^{18}, [1]^{27}, [3]^5, [5]^6, [45]^1$$

and

$$\text{sp}(S^+(U_H^3)) = \{-7-2i\}^{15}, [-7+2i]^{15}, [-5]^6, [-1]^{24}, [1]^{21}, [3]^2, [5]^9, [45]^1.$$

This means that we are able to distinguish the pair of SRGs using the spectrum of $S^+(U^3)$.

Consequently, we propose the use of the spectrum of the matrix representation $S^+(U_G^3)$ for a graph G for distinguishing it from other graphs that are cospectral with respect to standard matrix representations. The spectrum of $S^+(U_G^3)$ can be calculated in time $O(|\mathcal{E}_G|^3)$, thus a pair of graphs can be compared in time polynomial in the size of the graphs.

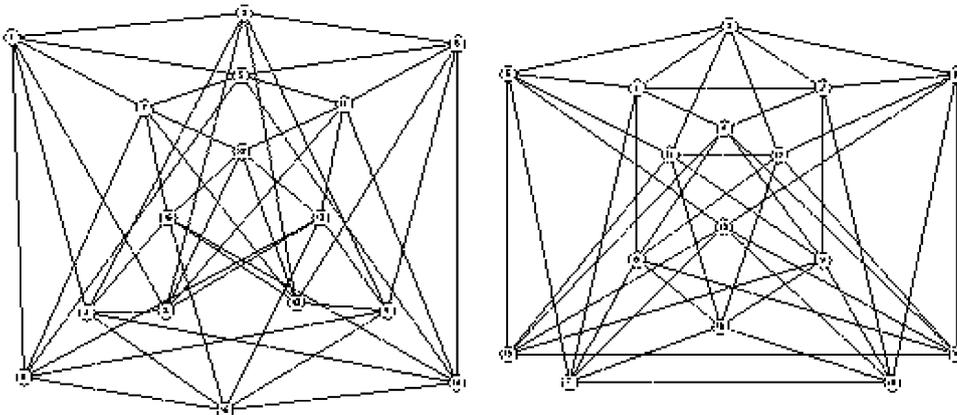


Fig. 4. Two non-isomorphic SRGs (*G* left, *H* right) with the parameter set (16, 6, 2, 2) (The graphs were drawn using Bill Kocay's 'Graphs and Groups' program available at <http://bkocay.cs.umanitoba.ca/G&G/G&G.html>).

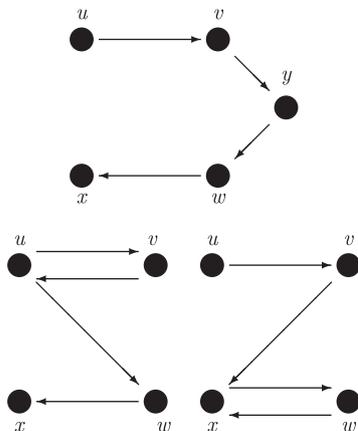


Fig. 5. The paths of length 3 between a pair of states (u, v) and (w, x) which are assigned a positive amplitude (above) and a negative amplitude (below) (u, v, w and w are all distinct). The top path is possible if there exists a vertex y such that $y \sim v$ and $y \sim w$, one of the bottom paths is possible if $u \sim w$ or $v \sim x$. Note that there will not be paths of both positive and negative amplitudes of length t between (u, v) and (w, x) if $t < 3$.

The third power is important because it is the lowest power for which destructive interference can take place between a general pair of states in the graph. That is, the Grover coins which are used to give the amplitudes for the entries of U , assign a positive quantum amplitude to transitions of the form $(u, v) \rightarrow (v, w)$, and negative quantum amplitudes to transitions of the form $(u, v) \rightarrow (v, u)$ (provided that $d(v) > 2$). The third power of U gives the sums of the amplitudes for the paths of length three between each pair of states. Consider a pair of states of the walk, (u, v) and (w, x) , where u, v, w and x are distinct vertices. The third power is the lowest power for which destructive interference takes place between the paths between a pair of states in general position in the graph (i.e. where all the vertices are distinct) as shown in Fig. 5. It is thus the lowest power for which the interference effects become important for all entries in U^t .

3.2.2. Constructing $S^+(U^3)$ for SRGs

For an SRG, G , with parameters (n, k, λ, μ) and adjacency matrix A , we can make use of the regularity of its structure to construct $S^+(U^3)$ directly using the conditions below. The development of this construction is given in [27]. The matrix $S^+(U_G^3)$ is such that $S^+(U_G^3)_{(u,v)(w,x)} = 1$ if and only if one of the following conditions

holds:

1. $u = x, v \neq w$ and

$$\mu + (\lambda - \mu)A_{vw} - k + \frac{k^2}{4} > 0$$

(which always holds if $u = x, v \neq w$ and $\mu, \lambda > 0$);

2. $u = w, x \neq v$ and $A_{vx} < 2\lambda/k$;
3. $u = w$ and $x = v$;
4. $u \neq w, x = v$ and $A_{uw} < 2\lambda/k$;
5. $u \neq w, u \neq x, v \neq w, v \neq x$ and

$$\frac{2}{k}(\mu + (\lambda - \mu)A_{vw}) > A_{uw} + A_{vx}.$$

Thus the matrix $S^+(U_G^3)$ for an SRG, G , can be constructed directly, without the need to first construct U and carry out matrix multiplication. The spectrum of $S^+(U_G^3)$ can then be used to represent the graph G , and this can be calculated in time $O(n^3 k^3)$. Note that a similar construction is not possible for trees (which we carry out experiments with later) since there are not parameters determining their regularity.

4. Experiments

As noted earlier, traditional spectral methods for graph analysis tasks, such as computing graph edit distance, clustering, and identifying vertex correspondences rely on the use of the spectrum of either the adjacency matrix or the Laplacian matrix. However, these methods fail when confronted with a pair of non-isomorphic but cospectral graphs. We wish to investigate whether the use of the spectrum of $S^+(U^3)$ will help us resolve the ambiguities that result from cospectrality. Firstly, we investigate how effective the spectrum of $S^+(U^3)$ is at distinguishing non-isomorphic graphs. Secondly, we investigate how well the distances between graphs calculated using the spectrum of $S^+(U^3)$ can be used to identify similar graphs derived from real-world data. We perform three sets of experiments.

Our first set of experiments uses sets of SRGs, bipartite incidence graphs of BIBDs, and 3-level regular graphs. Each of these sets of graphs have the property that every graph with the same set of parameters has the same adjacency matrix and Laplacian matrix spectra. What is more, distinguishing these graphs is, for each co-parametric set, a hard case of the graph isomorphism problem [29]. That is, there is no polynomial time algorithm proven to be able to distinguish between the graphs belonging to these sets.

Our second set of experiments investigates how often problems of cospectrality are likely to occur in real-world data represented by graphs. Regular graphs and trees are two types of graph that are commonly used to represent real-world data in pattern recognition contexts [30]. We therefore generate all 3,957,097 regular graphs on up to 14 vertices and all 63,242,255 trees on up to 24 vertices. Using these data, we investigate how many regular graphs and trees are cospectral with respect to the Laplacian matrix, the adjacency matrix, and the matrix $S^+(U^3)$.

Our third set of experiments uses graphs generated from two-dimensional images of three-dimensional objects. The images of the objects are from the COIL-100 database of 100 commonplace objects. The images were captured with a CCD camera with a 25 mm lens. The objects were placed on a turntable with a black background and ambient lighting. The objects were viewed with a camera depression angle of 25°. For each object 72 images were captured with camera fixed and the turntable rotating in angular increments of 5° [31]. The image resolution is 128 × 128 pixels. From each image in the sequence corner features are extracted using the Harris corner detector [32]. The extracted corner features are used as the seeds of the Voronoi tessellation of the image plane. The region adjacency graph of the Voronoi polygons, or Delaunay triangulation, is the graph structure used in our experiments. We calculate pairwise distances between these graphs using the Euclidean distances between their $S^+(U^3)$ spectra and perform multidimensional scaling (MDS) to visualise the distribution of graphs. We study the embedding of the graph in a two-dimensional plane produced by MDS to investigate how well the distances cluster the graphs.

4.1. Graph isomorphism testing for sets of cospectral graphs

To provide examples of graphs that can be considered ‘hard’ to distinguish, Mathon [29] proposes a hierarchy of sets of graphs that provide problems for testing graph isomorphism algorithms. These sets of graphs are SRGs (also known as non-transitive 2-level regular graphs), the balanced incidence graphs of non-symmetric BIBDs, and non-transitive 3-level regular graphs. In what follows, we describe the structure of these sets of graphs and investigate the performance of our algorithm at distinguishing non-isomorphic graphs belonging to them.

4.1.1. Strongly regular graphs

SRGs fall into a larger class of graphs referred to as t -level regular graphs. SRGs are those graphs that are exactly 2-level regular. Let $G=(\mathcal{V}, \mathcal{E})$ be a t -level regular graph and S_1 and S_2 be subsets of \mathcal{V} of at most t vertices. That G is t -level regular implies that if the induced subgraphs on S_1 and S_2 are isomorphic then the number of vertices adjacent to every vertex in S_1 is equal to the number of vertices that are adjacent to every vertex in S_2 [12]. Consider, for example, 1-level regular graphs. In this case the subsets are just single vertices and so the induced subgraphs are again just single vertices. Moreover, all 1-regular graphs are isomorphic. Therefore, to be 1-level regular, the condition is that every vertex is adjacent to the same number of vertices. Such graph are more commonly simply referred to as regular graphs.

As mentioned earlier, the 2-level regular graphs are SRGs. This is because there are two non-isomorphic subgraphs on two vertices. One of these is where the two vertices are adjacent and the other where they are not adjacent. Recall, from the definition of SRGs given in Section 3.2, that every pair of adjacent vertices share λ common neighbours and every pair of non-adjacent vertices share μ common neighbours. Thus we see that SRGs are indeed 2-level regular.

We have tested the effectiveness of the spectrum of $S^+(U^3)$ for distinguishing coparametric, but non-isomorphic SRGs. The parameters of the graphs we used, together with the number of graphs with

Table 1
The SRGs used to test the algorithm

(n, k, λ, μ)	Number of coparametric SRG
(16, 6, 2, 2)	2
(16, 9, 4, 6)	2
(25, 12, 5, 6)	15
(26, 10, 3, 4)	10
(28, 12, 6, 4)	4
(29, 14, 6, 7)	41
(35, 18, 9, 9)	227
(36, 14, 4, 6)	180
(36, 15, 6, 6)	32,548
(40, 12, 2, 4)	28
(45, 12, 3, 3)	78
(64, 18, 2, 6)	167

These SRGs were obtained from Ref. [33].

these parameters, are given in Table 1. For every SRG we found that the spectrum of $S^+(U^3)$ was distinct from those of all the other SRGs with the same parameters. Also, the differences between the spectra of two non-isomorphic graphs were always in excess of 10 orders of magnitude greater than the numerical accuracy used and so this never presented any problems when distinguishing them. Since SRGs with distinct parameters can be easily distinguished, and since the spectrum of $S^+(U^3)$ is invariant under permutations of the vertices, the algorithm was, therefore, able to solve the graph isomorphism problem for all the SRGs that we tested it on.

To characterise these results we compute a vector, \mathbf{s}_G , of the ordered eigenvalues of $S^+(U^3_G)$, for each graph G in one of the coparametric families. Let N be the number of graphs in the coparametric family. We define the distance between a graph k and a graph l to be $d_{k,l} = |\mathbf{s}_k - \mathbf{s}_l|$, for all graphs in the family. We construct the $N \times N$ distance matrix D whose (k, l) th entry is given by d_{kl} . We found that $d_{kl} = 0$ if and only if $k = l$, thus distinguishing all non-isomorphic graphs. As an example, the matrix for the SRG with parameters (26, 10, 3, 4) is

$$D = \begin{pmatrix} 0 & 4.13 & 42.88 & 26.64 & 22.90 & 26.21 & 45.13 & 26.11 & 23.54 & 23.36 \\ 4.13 & 0 & 45.49 & 25.43 & 22.30 & 24.60 & 51.95 & 29.34 & 24.85 & 23.79 \\ 42.88 & 45.49 & 0 & 53.42 & 55.58 & 58.84 & 15.50 & 96.27 & 53.68 & 57.49 \\ 26.64 & 25.43 & 53.42 & 0 & 3.08 & 3.86 & 53.24 & 75.14 & 3.63 & 3.06 \\ 22.90 & 22.30 & 55.58 & 3.08 & 0 & 2.46 & 53.46 & 68.05 & 2.49 & 1.17 \\ 26.21 & 24.60 & 58.84 & 3.860 & 2.46 & 0 & 57.21 & 71.88 & 3.38 & 2.53 \\ 45.13 & 51.95 & 15.50 & 53.24 & 53.46 & 57.21 & 0 & 94.33 & 51.90 & 55.51 \\ 26.11 & 29.34 & 96.27 & 75.14 & 68.05 & 71.88 & 94.33 & 0 & 71.37 & 68.36 \\ 23.54 & 24.85 & 53.68 & 3.63 & 2.49 & 3.38 & 51.90 & 71.37 & 0 & 1.89 \\ 23.36 & 23.79 & 57.49 & 3.06 & 1.17 & 2.53 & 55.51 & 68.36 & 1.89 & 0 \end{pmatrix}.$$

We visualise the distribution of the graphs given by the spectrum of $S^+(U^3)$ by performing principal components analysis (PCA) on the spectra of the graphs. For N graphs, each graph is represented by a vector \mathbf{s}_i ($1 \leq i \leq N$) of fixed length, c . Let $\bar{\mathbf{s}} = \sum_k \mathbf{s}_k / N$ be the mean vector for the dataset. We construct the matrix $X = [\mathbf{s}_1 - \bar{\mathbf{s}} | \mathbf{s}_2 - \bar{\mathbf{s}} | \dots | \mathbf{s}_N - \bar{\mathbf{s}}]$ and then compute the covariance matrix for the set of vectors by taking the matrix product $C = XX^T$. We extract the principal component vectors by performing the eigendecomposition $C = (1/N) \sum_{j=1}^N l_j \mathbf{u}_j \mathbf{u}_j^T$ on the covariance matrix C , where the l_i are the eigenvalues and the \mathbf{u}_i are the eigenvectors. The eigenvectors and the associated eigenvalues are ordered according to the size of the eigenvalues, with the largest eigenvalue first.

We use the first r eigenvectors (two or three in practice for visualisation purposes) to represent the graphs. The coordinate system of the eigenspace is spanned by the z orthogonal vectors $B = (\mathbf{b}_1 | \mathbf{b}_2 | \dots | \mathbf{b}_z)$. The individual graphs represented by the vectors \mathbf{s}_k are projected onto this eigenspace using the formula $\mathbf{y}_k = B^T(\mathbf{s}_k - \bar{\mathbf{s}})$. Hence each graph is represented by a z component vector \mathbf{y}_k in the eigenspace.

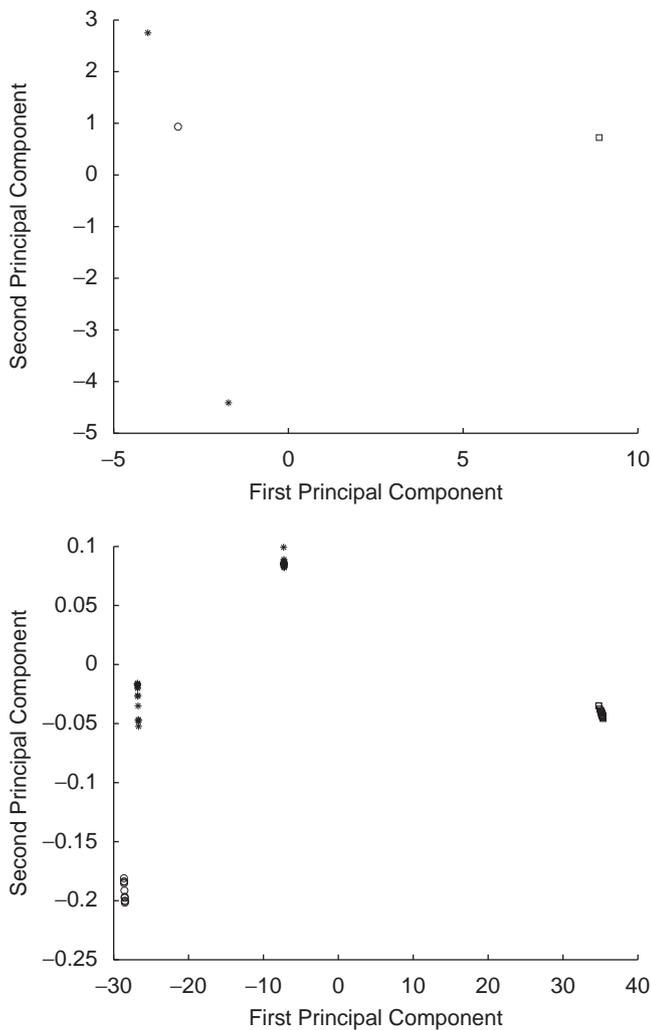


Fig. 6. PCA embeddings of SRG with distances calculated using the spectra of A (top) and the spectra of $S^+(U^3)$ (bottom). The sets are those with parameters (25, 12, 5, 6), plotted as a *; (26, 10, 3, 4), plotted as a o; (29, 14, 6, 7), plotted as a +; and (40, 12, 2, 4), plotted as a □.

Using PCA we embed the sets of graphs with parameters (25, 12, 5, 6), (26, 10, 3, 4), (29, 14, 6, 7) and (40, 12, 2, 4) in two-dimensional space. Since the graphs are of different sizes and hence have different numbers of eigenvalues, we order the eigenvalues and create vectors for each graph of length $c = 20$ by taking the largest 20 eigenvalues to represent each graph. Cospectral pairs of graphs, i.e. those of zero distance, will appear as coincident points under the embedding. If a pair of graphs are not coincident in the embedding then they are not cospectral. Fig. 6 shows two embeddings of the four sets of coparametric SRGs. The first is obtained using distances calculated from the spectrum of their adjacency matrix and the second is obtained using distances from the spectra of $S^+(U^3)$. Although the embedding using the spectrum of the adjacency matrix separates the individual sets, all graphs with the same set of parameters are mapped to the same point indicating that the graphs are not distinguished using this method. The embedding using the spectra of $S^+(U^3)$, on the other hand, distinguishes all the graphs within each set and thus no two graphs are coincident in the embedding. We can also see from the plot that the first principal component, the x -component, distinguishes the individual sets and the second principal component, the y -component, distinguishes the individual graphs in the sets. The variation represented

Table 2

The parameters (v, k, λ) of the sets of balanced incomplete block designs (BIBD) represented by the graphs and the number of pairs of graphs from each set that are cospectral when $S^+(U^3)$ is used

Parameters	Number of graphs	Number of pairs	Number of cospectral pairs using $S^+(U^3)$
(15, 3, 1) ^a	80	3160	476
(16, 6, 2) ^b	3	3	3
(23, 11, 5) ^b	1106	611,065	524
(25, 9, 3) ^b	78	3003	28
(31, 10, 3) ^b	151	11,325	69
(37, 9, 2) ^a	3	3	3
(56, 11, 2) ^a	3	3	3
(61, 25, 10) ^b	18	153	0

Note, it is more appropriate to consider the number of cospectral pairs rather than number of graphs with cospectral partners since, for each set, every graph is cospectral with every other graph when using the adjacency or Laplacian matrices. The graphs marked (a) were obtained from Ref. [33], those marked (b) were obtained from Ref. [37].

by the first principal component is also significantly greater than that recorded by the second principal component. This is as we would expect, since the difference between graphs from different parameter sets is greater than the difference between graphs with the same parameters. Thus the PCA embedding is effective at using the distances to distinguish variations within the coparametric sets from variations between the sets.

4.1.2. Bipartite incidence graphs of non-symmetric BIBDs

BIBDs are used for, amongst other things, experimental design, error-correcting codes and cryptography [34,35]. Colbourn and Colbourn [36] showed that an arbitrary graph can be encoded uniquely as a BIBD and thus provides evidence for the conjecture that isomorphism testing of block designs is a hard subcase of the graph isomorphism problem.

A BIBD with parameters (v, k, λ) is a set of $b = v(v - 1)\lambda/k(k - 1)$ blocks, Γ , each containing k points from a set, Ω , where $|\Omega| = v$. The design is such that each 2-subset from Ω occurs in λ blocks. A BIBD is represented graphically as a bipartite incidence graph with vertex set $\Omega \cup \Gamma$ and adjacency matrix

$$A_{jk} = \begin{cases} 1 & \text{if } j \in \Omega \text{ and } j \in k \in \Gamma \text{ or } k \in \Omega \text{ and } k \in j \in \Gamma, \\ 0 & \text{otherwise.} \end{cases}$$

That is, vertices represent either blocks or points and the vertex for point j is adjacent to the vertex for block k if $j \in k$.

All BIBDs with the same parameter set are cospectral with respect to their adjacency matrix and Laplacian matrix representations. We have tested the effectiveness of the spectrum of $S^+(U^3)$ for distinguishing coparametric BIBDs. The parameters of the BIBDs used, the number of BIBDs with these parameters, and the number of pairs that are cospectral using $S^+(U^3)$ are given in Table 2. In Ref. [27] we give a set of conditions for the existence of non-zero entries in $S^+(U^3)$ for BIBDs as we did for SRGs in Section 3.2.2. However, we show that such conditions must include terms that are quadratic in the entries of A rather than just linear terms which were sufficient for SRGs.

Unlike the SRGs, some of the BIBDs cannot be distinguished using the spectrum of $S^+(U^3)$. For the sets with parameters (16, 6, 2), (37, 9, 2) and (56, 11, 2) the spectrum of $S^+(U^3)$ was unable to distinguish any of the three graphs from the other two graphs with the same parameters. For the BIBDs with parameters (15, 3, 1), (23, 11, 5), (25, 9, 3) and (31, 10, 3) on the other hand, we were able to distinguish between 85% and 99.91% of all the pairs in each set. Meanwhile, for the set with parameters (61, 25, 10) we were able to distinguish between every BIBD in the set.

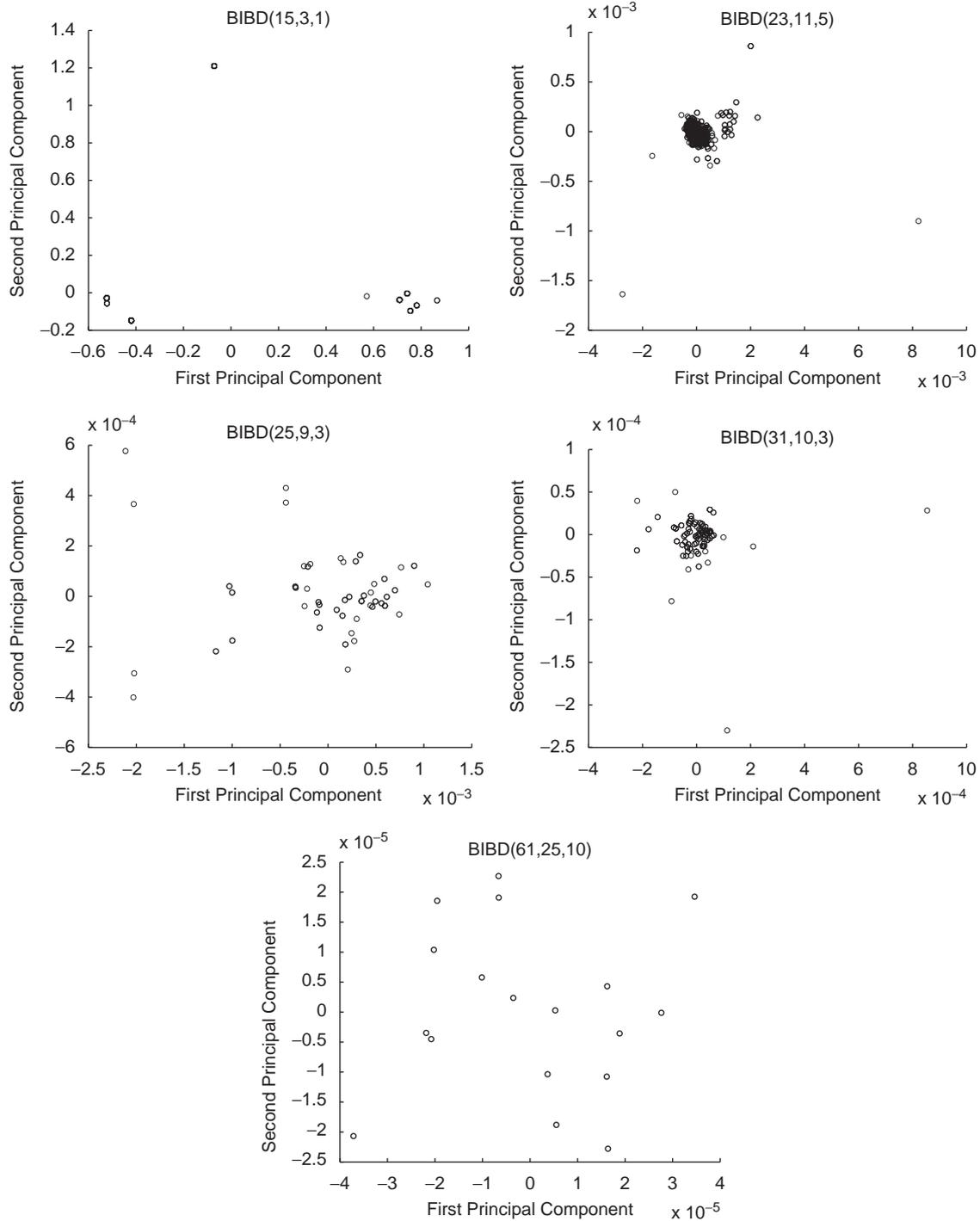


Fig. 7. PCA embeddings of the sets of bipartite incidence graphs of the BIBD with parameters (15, 3, 1), (23, 11, 5), (25, 9, 3), (31, 10, 3) and (61, 25, 10). Note that all these graphs are cospectral with respect to their adjacency and Laplacian matrices.

The sets of graphs with parameters (15, 3, 1), (23, 11, 5), (25, 9, 3), (31, 10, 3) and (61, 25, 10) have been embedded using the procedure described for SRGs and these embeddings are shown in Fig. 7. The sets with parameters (16, 6, 2), (37, 9, 2) and (56, 11, 2) have been embedded together in Fig. 8. These sets cannot be embedded separately since the covariance matrix for the graphs in a particular set is the all-zero matrix on which it is not possible to perform eigendecomposition. In Fig. 7 the graphs have been distributed across the plane, and this indicates that the majority of the graphs are distinguished

from one another. The apparent clustering of the graphs from the sets with parameters (23, 11, 5) and (31, 10, 3) is due to the large number of graphs in these sets, 1106 and 151, respectively. All but 0.6% of the pairs in the set with parameters (31, 10, 3) and 0.09% of the pairs in the set with parameters (23, 11, 5) are distinguished using the spectrum of $S^+(U^3)$. The sets with parameters (16, 6, 2), (37, 9, 2) and (56, 11, 2) in Fig. 8 have all been distinguished from one another. However, the graphs in a particular set are all coincident. This indicates that the spectral distance between these graphs is zero when

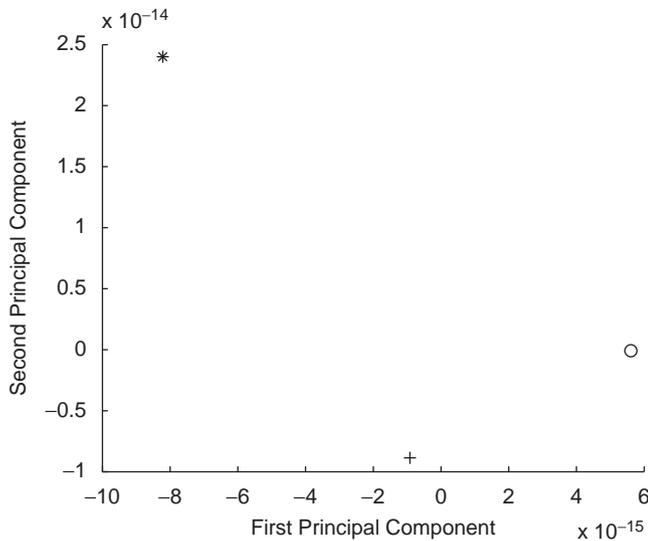


Fig. 8. PCA embedding of the sets of bipartite incidence graphs of the BIBD with parameters $(16, 6, 2)$ embedded as a *, $(37, 9, 2)$ embedded as a o and $(56, 11, 2)$ embedded as a +.

$S^+(U^3)$ is used, as is also the case with the adjacency spectrum and Laplacian spectrum.

4.1.3. 3-Level regular graphs

As mentioned earlier, a regular graph is classed as 1-level regular and an SRG as 2-level regular. Next in this hierarchy are the 3-level regular graphs. Due to their high level of symmetry these graphs are very difficult to distinguish [38]. We use the method given by Mathon [29] for constructing a 3-level regular graph given an SRG with parameters $(n, 2\mu, \lambda, \mu)$. For the SRGs listed in Table 1, we construct all the possible 3-level regular graphs. Thus, using the sets with parameters $(25, 12, 5, 6)$, $(29, 14, 6, 7)$ and $(35, 18, 9, 9)$ we construct 3-level regular graphs on 52, 60 and 72 vertices, respectively. Using the spectra of $S^+(U^3)$ we were able to distinguish between all these graphs.

The PCA embeddings of the graphs using the distances calculated from their spectra are shown in Fig. 9. The 3-level regular graphs on 72 vertices appear to be clustered uniformly other than a few outliers. For the sets on 52 and 60 vertices, however, the embedding clusters the graphs into a number of distinct sets. In the embeddings, the first principal component distinguishes between the clusters and the second principal component records the variation within the clusters. It would be interesting to investigate whether this is a salient property of these sets of graphs.

Thus, in the hierarchy of graphs which pose problems for graph isomorphism testing, the spectra of $S^+(U^3)$ is able to distinguish between all SRGs (2-level regular graphs) that we have tested, all 3-level regular graphs, and the overwhelming majority of the bipartite incidence graphs of BIBDs. This demonstrates that for graphs that have high levels of symmetry and which provide some of the hardest classes for graph isomorphism testing the spectrum of $S^+(U^3)$ is able to correctly distinguish between non-isomorphic graphs. In the next section we consider smaller regular graphs and trees so as to investigate how common the problem of cospectrality is, and how successfully it is dealt with in these cases by using the spectrum of $S^+(U^3)$.

4.2. Cospectrality of regular graphs and trees

The cospectrality of graphs with respect to standard matrix representations is not confined to those with specific symmetries such

as SRGs. Trees are used frequently to represent structures to be compared [30] and often this is done using their spectra [39]. Regular graphs also occur if, for example, the structure encodes the k -nearest neighbours of a set of points. In this section we investigate how common the problem of cospectrality is for such graphs and how successfully the spectrum of $S^+(U^3)$ overcomes this problem. We investigate how often cospectrality arises using both the standard matrix representations and $S^+(U^3)$ for all regular graphs with up to 14 vertices and on all trees with up to 24 vertices.

4.2.1. Regular graphs

In Ref. [40], Zhu and Wilson investigate the number of trees with up to 21 vertices that are cospectral with respect to a number of standard matrix representations. They found that of these matrix representations, the spectrum of the Laplacian of the graphs performed best at distinguishing regular graphs. We extend this study to include the use of the spectrum of $S^+(U^3)$ for all regular graphs up to, and including, the 5-regular graphs on 14 vertices. The results are shown in Table 3. We can see that a large proportion of the regular graphs have a cospectral partner if the Laplacian matrix is used and that this proportion rises to as high as 34% for the 6-regular graphs on 10 vertices. The use of $S^+(U^3)$, on the other hand, far outperforms this. For all but the 4-regular graphs on 14 vertices, not a single graph has a cospectral partner. For these 88,168 4-regular graphs on 14 vertices just 10 have a cospectral partner when $S^+(U^3)$ is used compared with 11,165 when the Laplacian is used. Whilst cospectrality was a common problem when the Laplacian matrix was used, only 10 of the 3,957,097 regular graphs we tested were cospectral with respect to $S^+(U^3)$.

4.2.2. Trees

Trees are particularly problematic when spectral analysis is attempted. Schwenk [13] showed that almost all trees on sufficiently many vertices will have a partner that is cospectral with respect to their adjacency matrices. Merris went further than this by showing that the same is also true if the Laplacian matrix, adjacency matrix or any of their generalisations are used [14]. For all trees on up to 24 vertices, we have investigated how many are cospectral with respect to their Laplacian and $S^+(U^3)$ matrix representations (we found that the Laplacian spectrum was always better than the adjacency matrix spectra for distinguishing the trees). The results are shown in Table 4. The smallest trees for which the problem of cospectrality arises when the Laplacian is used have 11 vertices. Using $S^+(U^3)$, on the other hand, no two trees with fewer than 19 vertices share the same spectrum. For this set, only two of the 317,955 trees are cospectral when $S^+(U^3)$ is used, compared with 648 when the Laplacian is used. For the trees with a larger number of vertices, the number of trees that have a cospectral partner with respect to the Laplacian matrix is, at the least, 10 times greater than the number that are cospectral with respect to $S^+(U^3)$. Thus, the spectrum of $S^+(U^3)$ performs far better at distinguishing both regular graphs and trees on small numbers of vertices. We see no reason why this should not be the case for graphs on more vertices.

4.3. Clustering Delaunay triangulations from images

We have seen that the spectrum of $S^+(U^3)$ is able to distinguish between sets of graphs that form comparatively hard subcases of the graph isomorphism problem. What is more, the spectrum of $S^+(U^3)$ distinguishes between a far greater number of the regular graphs and trees that we tested than the Laplacian spectrum does. We now consider the use of the spectrum of $S^+(U^3)$ for clustering graphs representing images from the COIL-100 database of views of common-place objects. For each object in the COIL database there

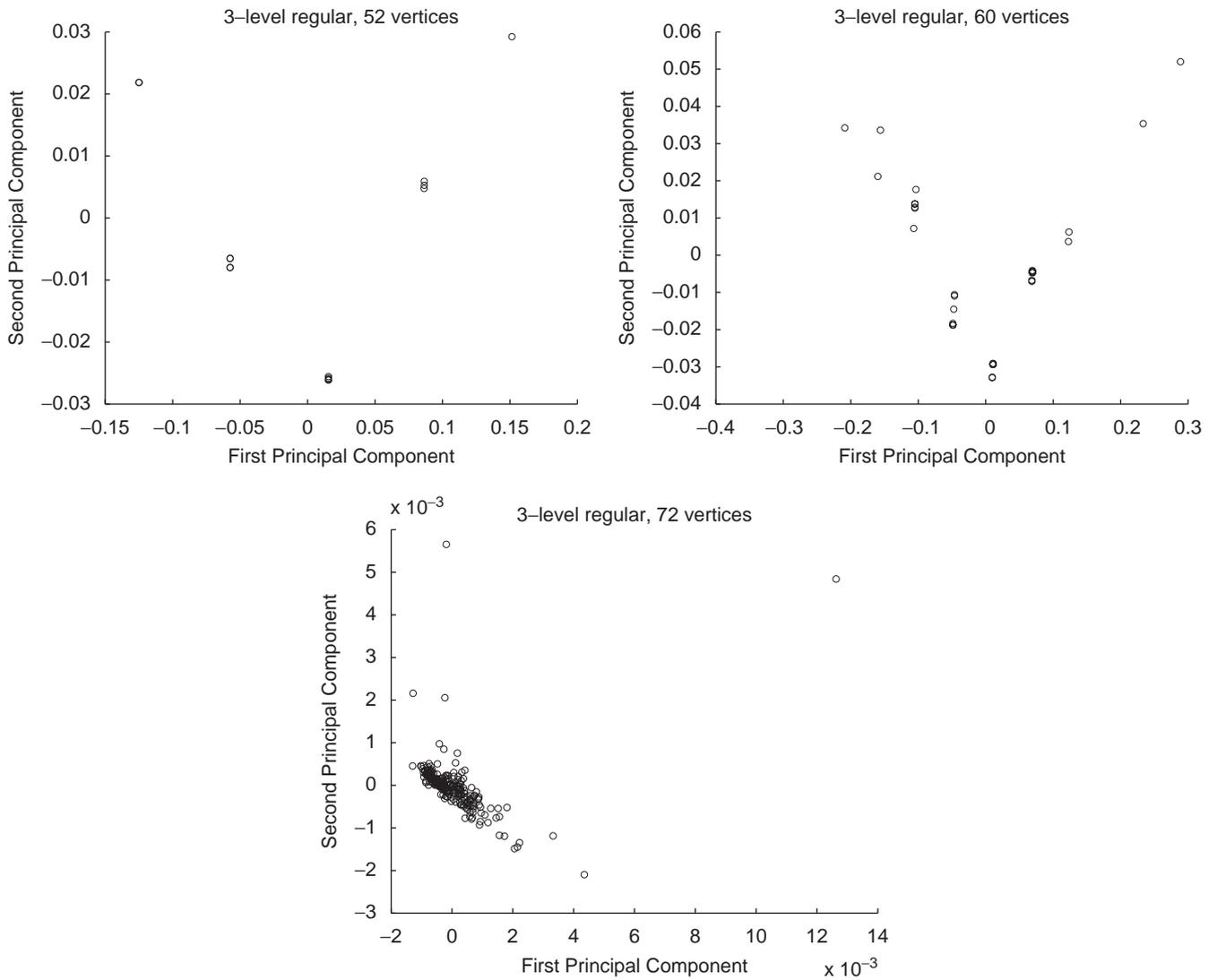


Fig. 9. PCA embeddings of the sets of 3-level regular graphs on 52, 60 and 72 vertices derived from the SRGs with parameters (25, 12, 5, 6), (29, 14, 6, 7) and (35, 18, 9, 9), respectively. Note that within each set, all the graphs are cospectral with respect to their adjacency and Laplacian matrices.

Table 3

The number of k -regular graphs on $|V|$ vertices ($|V| \geq 10$) that have a cospectral partner with respect to the Laplacian matrix, L , and the number that have a cospectral partner with respect to $S^+(U^3)$

$ V $	k	Number of graphs	Number cospectral w.r.t L	Number cospectral w.r.t $S^+(U^3)$
10	4	59	4	0
10	5	60	4	0
10	7	5	0	0
11	4	265	28	0
11	6	266	28	0
12	3	85	0	0
12	4	1544	258	0
12	5	7848	2689	0
12	6	7849	2689	0
12	7	1547	258	0
12	8	94	0	0
13	4	10,778	1753	0
13	6	367,860	98,076	0
13	8	10,786	1753	0
14	3	509	6	0
14	4	88,168	11,165	10
14	5	3,459,383	692,993	0

Table 4

The number of trees on $|V|$ vertices that have a cospectral partner with respect to the Laplacian, L , and $S^+(U^3)$

$ V $	Number of trees	Cospectral L	Cospectral $S^+(U^3)$
≤ 10	200	0	0
11	235	6	0
12	551	6	0
13	1301	18	0
14	3159	30	0
15	7741	48	0
16	19,320	68	0
17	48,629	221	0
18	123,867	230	0
19	317,955	440	2
20	823,065	648	2
21	2,144,505	1056	24
22	5,623,756	1563	32
23	14,828,074	2858	68
24	39,299,897	3623	290

are 72 views with camera viewing angle equally spaced around the object in 5° intervals.

We use nine views of each object, in 10° intervals to give a set, Ω , of 45 graphs. For a given image of an object we extract corner

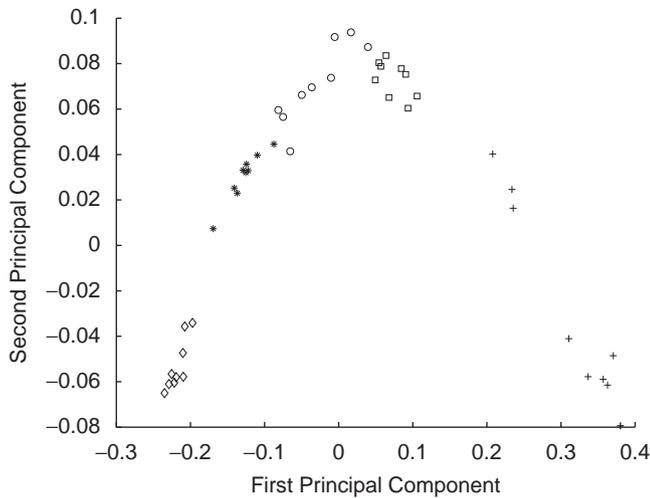


Fig. 10. The objects from the COIL database embedded using MDS on the spectrum of $S^+(U^3)$ of the Delaunay graphs. White cup +, earthenware cup o, cat toy *, wooden shape \square and car \diamond .

features using the Harris corner detector [32]. The corner points are used to seed a Voronoi tessellation of the image plane. Our graph representation is based on the region adjacency matrix for the Voronoi regions, i.e. the Delaunay triangulation of the detected corner features. The process of constructing the graph from an image is illustrated in Fig. 12.

For each graph, G , we compute a column vector, \mathbf{s}_G , of the ordered eigenvalues of $S^+(U_G^3)$. As the graphs are of different sizes and thus their spectra are of different lengths, the vectors are all made to be the same length by appending zeros to the ends of the shorter vectors. The distance matrix, D , with (i, j) entry given by $d_{ij} = \|\mathbf{s}_i - \mathbf{s}_j\|$, where $\|\cdot\|$ is the standard Euclidean norm. The graphs are then embedded in a two-dimensional space by carrying out MDS on this matrix. MDS is used in preference to PCA since the complexity of PCA scales with the size of the vector description for the graphs. The complexity of MDS on the other hand scales with the number of graphs compared. What is more, PCA and MDS are equivalent when Euclidean distances are used to calculate the distance matrix used for MDS [41].

MDS is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. Let N be the number of graphs. The first step of MDS is to calculate a matrix, F , whose element with row r and column c is given by $F_{rc} = -\frac{1}{2}(d_{rc}^2 - \hat{d}_r^2 - \hat{d}_c^2 + \hat{d}_{..}^2)$, where $\hat{d}_r = (1/N)\sum_{c=1}^N d_{rc}$ is the average distance over the r th row, \hat{d}_c is the average distance over the c th column and $\hat{d}_{..} = (1/N^2)\sum_{r=1}^N \sum_{c=1}^N d_{rc}$ is the average distance over all rows and columns of the distance matrix d .

We subject the matrix F to an eigenvector analysis to obtain a matrix of embedding coordinates Y . If the rank of F is k , where $k < N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, l_1, l_2, \dots, l_k , where $l_1 \geq l_2 \geq \dots \geq l_k \geq 0$. The corresponding ordered eigenvectors are denoted by \mathbf{u}_j ($1 \leq j \leq k$). The embedding coordinate system for the graphs is $Y = [\sqrt{l_1}\mathbf{u}_1, \sqrt{l_2}\mathbf{u}_2, \dots, \sqrt{l_k}\mathbf{u}_k]$. For the graph indexed r , the vector of embedding coordinates, \mathbf{y}_r , is a row of matrix Y , given by $\mathbf{y}_r = (Y_{r,1}, Y_{r,2}, \dots, Y_{r,k})^T$.

The embedding of the graphs using MDS on the distances between the spectrum of $S^+(U^3)$ is shown in Fig. 10. For comparison, the MDS embedding obtained using the Laplacian spectrum is shown in Fig. 11. The ability of both approaches to cluster the items is comparable. The graphs in both embeddings appear to be restricted to a one-dimensional curve. However, this is a less noticeable prob-

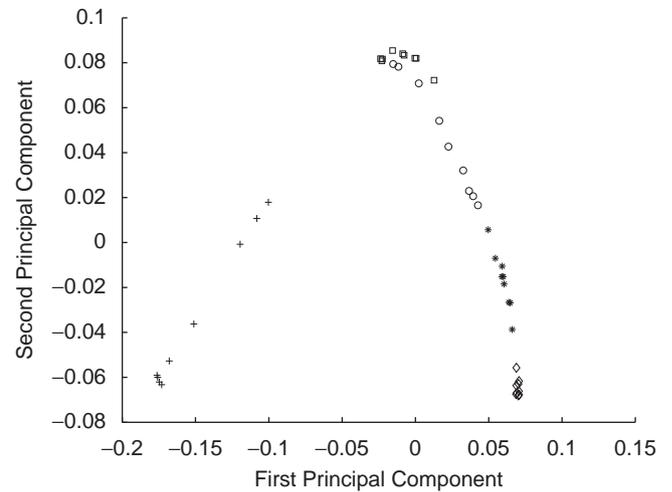


Fig. 11. The objects from the COIL database embedded using MDS on the spectrum of the Laplacian of the Delaunay graphs. White cup +, earthenware cup o, cat toy *, wooden shape \square and car \diamond .

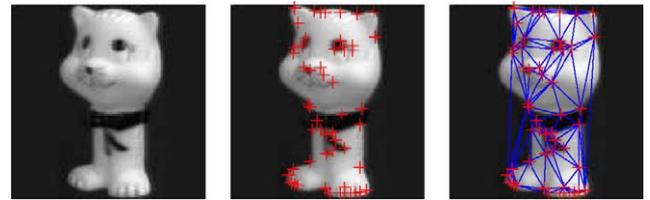


Fig. 12. The original image (left), the extracted features (middle), and the Delaunay triangulation (right) for a sample object.

lem when using the spectrum of $S^+(U^3)$ than when using the Laplacian spectrum. We can see from the embeddings that the spectrum of $S^+(U^3)$ is also more effective at distinguishing the earthenware cup (o) from the wooden shape (\square) than the Laplacian spectrum is. The clusters of these two objects are not as clearly separated in the Laplacian embedding as in the $S^+(U^3)$ embedding. By manually drawing cluster boundaries in the MDS plots, in the case of the spectrum of $S^+(U^3)$ there is one error while for the Laplacian there are three errors for the 45 COIL graphs. Thus we can see that we have a matrix representation of graphs, less prone to cospectrality than the standard adjacency matrix and Laplacian matrix representations. The spectrum of this matrix can also be used effectively to cluster graphs representing views of different common-place objects.

5. Conclusions

In this paper we have shown how a matrix representation, which we refer to as $S^+(U^3)$, can be used to lift the cospectrality often encountered when using the adjacency matrix or Laplacian matrix representations of a graph. Our representation is based on the sum of the quantum amplitudes of paths of the discrete-time quantum walk on the graph. Since quantum amplitudes are not restricted to positive real values, the representation depends crucially on the interference of different paths between pairs of states. We use the spectrum of $S^+(U^3)$ to distinguish between graphs which present the same adjacency and Laplacian spectra. The spectrum of $S^+(U^3)$ can be calculated in time $O(|\mathcal{E}|^3)$. For inexact graph matching we represent a graph by the vector of ordered eigenvalues of $S^+(U^3)$ and calculate the distance between graphs using the Euclidean distance between these vectors.

We have carried out three sets of experiments on the spectrum of $S^+(U^3)$. The first tested the ability of the spectrum to distinguish between graphs from sets of highly regular graphs. We considered SRGs (2-level regular graphs), bipartite incidence graphs of BIBDs and 3-level regular graphs. We found that the spectrum of $S^+(U^3)$ was able to distinguish between all SRGs and all 3-level regular graphs that we considered. It was also able to distinguish between the vast majority of bipartite incidence graphs of BIBDs. For all these classes of graphs, coparametric sets all present the same adjacency matrix and Laplacian matrix spectra and thus cannot be distinguished using the spectra of either of these matrices. What is more, there is no polynomial time algorithm proven to be able to distinguish between coparametric graphs from any of these sets.

The second set of experiments investigates how frequently regular graphs and trees are cospectral when the Laplacian matrix is used compared to when $S^+(U^3)$ is used. Such graphs provide examples of those that may be encountered in real-world data applications. Whereas cospectrality was common when using the Laplacian spectrum, the proportion of graphs which had a cospectral partner when $S^+(U^3)$ was used was significantly less. Of the 63,242,255 trees that we tested, we found that 10,815 of them had a cospectral partner when the Laplacian matrix was used and that only 418 had a cospectral partner when $S^+(U^3)$ was used. Similarly, of the 3,957,106 regular graphs tested, 811,704 had a cospectral partner with respect to the Laplacian matrix and 10 with respect to $S^+(U^3)$. Thus the method is far more robust against the problem of cospectrality than the adjacency or Laplacian spectra, on which many spectral methods are based. The third set of experiments uses distances calculated from the spectrum of $S^+(U^3)$ to cluster graphs derived from images of everyday objects. We found that the distances could be used to effectively cluster together different views of the same object, and separate the different objects. Thus, the spectrum of $S^+(U^3)$ is able to lift the cospectrality of many graphs, and distances calculated using its spectrum are able to reflect the dissimilarity of graphs derived from real-world data.

References

- [1] L. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the 28th Annual ACM Symposium on the Theory of Computation, New York, NY, ACM Press, New York, 1996, pp. 212–219.
- [2] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* 26 (1997) 1484–1509.
- [3] M.A. Nielsen, I.L. Chuang, *Quantum Computing and Quantum Information*, Cambridge University Press, Cambridge, 2000.
- [4] S.L. Braunstein, Quantum teleportation without irreversible detection, *Phys. Rev. A* (1996) 1900–1903.
- [5] A. Robles-Kelly, E.R. Hancock, Edit distance from graph spectra, in: Proceedings of the IEEE International Conference on Computer Vision, 2003, pp. 234–241.
- [6] L. Lovász, *Random Walks on Graphs: A Survey*, Combinatorics, Paul Erdős is Eighty, vol. 2, János Bolyai Mathematical Society, Budapest, 1996, pp. 353–398.
- [7] A. Sinclair, *Algorithms for Random Generation and Counting: A Markov Chain Approach*, Birkhauser Verlag, Boston, 1993.
- [8] P. Doyle, J. Snell, *Random Walks and Electrical Networks*, Mathematical Association of America, Washington, 1984.
- [9] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, *Comput. Networks ISDN Syst.* 30 (1–7) (1998) 107–117.
- [10] A. Robles-Kelly, E.R. Hancock, String edit distance, random walks and graph matching, *Int. J. Pattern Recognition Artif. Intell.* 18 (3) (2004) 315–327.
- [11] L. Sarti, M. Maggini, M. Gori, Exact and approximate graph matching using random walks, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (7) (2005) 1100–1111.
- [12] P.J. Cameron, Strongly regular graphs, in: *Topics in Algebraic Graph Theory*, Cambridge University Press, Cambridge, 2004, pp. 203–221.
- [13] A.J. Schwenk, Almost all trees are cospectral, in: F. Harary (Ed.), *New Directions in the Theory of Graphs*, Academic Press, New York, 1973, pp. 275–307.
- [14] R. Merris, Almost all trees are coisomorphic, *Linear Algebra Appl.* 150 (1991) 61–66.
- [15] J. Kempe, Quantum random walks—an introductory overview, *Contemp. Phys.* 44 (4) (2003) 307–327.
- [16] A. Ambainis, Quantum walks and their algorithmic applications, *Int. J. Quantum Inf.* 1 (2003) 507–518.
- [17] N. Shenvi, J. Kempe, K.B. Whaley, A quantum random walk search algorithm, *Phys. Rev. A* 67 (5) (2003).
- [18] A. Childs, E. Farhi, S. Gutmann, An example of the difference between quantum and classical random walks, *Quantum Inf. Process.* 1 (2002) 35.
- [19] J. Kempe, Quantum random walks hit exponentially faster, in: *Proceedings of the 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'03)*, 2003, pp. 354–369.
- [20] A. Ambainis, Quantum walk algorithm for element distinctness, in: *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, 2004.
- [21] P. Kwiat, J.R. Mitchell, P.D.D. Schwindt, A.G. White, Grover's search algorithm: an optical approach, *J. Mod. Opt.* 47 (2000) 257.
- [22] J. Cirac, P. Zoller, H.J. Kimble, H. Mabuchi, Quantum state transfer and entanglement distribution among distant nodes in a quantum network, *Phys. Rev. Lett.* 78 (1997) 3221–3224.
- [23] J.R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, July 1998.
- [24] D. Aharonov, A. Ambainis, J. Kempe, U. Vazirani, Quantum walks on graphs, in: *STOC '01: Proceedings of ACM Theory of Computing*, ACM Press, New York, 2001, pp. 50–59.
- [25] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, J. Watrous, One-dimensional quantum walks, in: *Proceedings of 33th STOC*, New York, NY, ACM, New York, 2001, pp. 60–69.
- [26] A. Nayak, A. Vishwanath, Quantum walk on a line, *DIMACS Technical Report 2000-43*, 2000.
- [27] D.M. Emms, *Analysis of graph structure using quantum walks*, Ph.D. Thesis, University of York, 2008.
- [28] S. Severini, On the structure of the adjacency matrix of the line digraph of a regular digraph, *Discrete Appl. Math.* 154 (12) (2006) 1663–1665.
- [29] R. MATHON, Sample graphs for isomorphism testing, in: *9th Southeastern Conference on Combinatorics*, vol. 21, 1978, pp. 499–517.
- [30] A. Torsello, D. Hidovic-Rowe, M. Pelillo, Polynomial-time metrics for attributed trees, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (7) (2005) 1087–1099.
- [31] S. Nene, S. Nayar, H. Murase, *Columbia object image library: COIL*, 1996.
- [32] C. Harris, M. Stephens, A combined corner and edge detector, in: *Proceedings of 4th Alvey Vision Conference*, Manchester, vol. 15, 1988, pp. 147–151.
- [33] E. Spence, Strongly regular graphs, 2006 (<http://www.maths.gla.ac.uk/es/srgraphs.html>).
- [34] C.J. Colbourn, J.H. Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, New York, 1996.
- [35] C.J. Colbourn, P.C. van Oorschot, Applications of combinatorial designs in computer science, *ACM Comput. Surv.* 21 (2) (1989) 223–250.
- [36] M.J. Colbourn, C.J. Colbourn, Concerning the complexity of deciding isomorphism of block designs, *Discrete Appl. Math.* 3 (1981) 155–162.
- [37] G. Royle, Strongly Regular Graphs, 2006 (<http://www.csse.uwa.edu.au/gordon/remote/designs/index.html>).
- [38] D. Spielman, Faster isomorphism testing of strongly regular graphs, in: *STOC '96: Proceedings of 28th ACM Symposium on Theory of Computing*, New York, NY, USA, ACM Press, New York, 1996, pp. 576–584.
- [39] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, S. Zucker, Indexing hierarchical structures using graph spectra, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (7) (July 2005).
- [40] P. Zhu, R. Wilson, A study of graph spectra for comparing graphs, in: *Proceedings of the 16th British Machine Vision Conference*, Oxford, 2005, pp. 499–517.
- [41] A. Webb, *Statistical Pattern Recognition*, second ed., Wiley, Chichester, England, 2002.

About the Author—DAVID EMMS received a first class honours degree in mathematical physics from the University of Nottingham in 2003. He completed the PhD degree at the University of York in 2007, under the supervision of Edwin Hancock. His research interests include quantum computing, structural pattern recognition and machine learning.

About the Author—SIMONE SEVERINI graduated from 'Liceo Classico Angelo Poliziano', and then studied Chemistry at the University of Siena and Philosophy at the University of Florence. In 2003 he obtained his PhD degree from Bristol University where he was supervised by Richard Jozsa. Following this he was at the University of York (England) with the Department of Mathematics and the Department of Computer Science between 2003 and 2007. He is currently a Postdoctoral Research Fellow at the Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo. He has also been a long term visitor at MSRI, MIT, Institut Poincaré, Newton Institute, CRI, CWI, Nankai. His research interests are discrete mathematical models of physical systems, their representations and dynamics mainly in quantum mechanics. He has published about 50 papers in scientific journals and conferences.

About the Author—EDWIN HANCOCK received the BSc degree in physics, with honors, in 1977 and the PhD degree in the area of high-energy physics in 1981 from the University of Durham. Following this, he worked for 10 years as researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period, he also held adjunct teaching posts at the University of Surrey

and the Open University. In 1991, he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to senior lecturer in 1997 and to reader in 1998. In 1998, he was appointed to as chair of Computer Vision. He now leads a group of some 20 faculty, research staff, and PhD students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimization and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph matching, shape-from-X, image databases, and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing, and medical imaging. He has published about 120 journal papers and 450 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 and an outstanding paper award in 1997 by the journal *Pattern Recognition*. In 1998, he became a fellow of the International Association for Pattern Recognition. He is also a fellow of the Institute of Physics, the Institute of Engineering and Technology, and the British Computer Society. He has been a member of the editorial boards of the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*. He has also been a guest editor for special editions of the journals *Image and Vision Computing* and *Pattern Recognition*. In 2006, he was appointed as the founding editor-in-chief of the *IET Computer Vision Journal*, and in 2008 Area Editor for *Computer Vision and Image Understanding*. He has been on the program committees for numerous national and international meetings. In 1997, with Marcello Pelillo, he established a new series of international meetings on energy minimization methods in computer vision and pattern recognition. In 2008 the University of Durham awarded him a DSc degree for his published research.

About the Author—RICHARD WILSON received the B.A. degree in physics from the University of Oxford in 1992. In 1996, he completed a PhD degree at the University of York in the area of pattern recognition. From 1996 to 1998, he worked as a Research Associate at the University of York. After a period of postdoctoral research, he was awarded an Advanced Research Fellowship in 1998. In 2003, he took up a lecturing post and he is now a Reader in the Department of Computer Science at the University of York. He has published some 100 papers in journals, edited books and refereed conferences. He received an outstanding paper award in the 1997 Pattern Recognition Society awards and has won the best paper prize in ACCV 2002. He is currently an Associate Editor of the journal *Pattern Recognition*. His research interests are in statistical and structural pattern recognition, graph methods for computer vision, high-level vision and scene understanding. He is a member of the IEEE Computer Society.