Quantum Information and Computation, Vol. 9, No. 3&4 (2009) 0231–0254

GRAPH EMBEDDING USING A QUASI-QUANTUM ANALOGUE OF THE HITTING TIMES OF CONTINUOUS TIME QUANTUM WALKS

DAVID M. EMMS
Department of Computer Science, University of York
York, YO10 5DD, United Kingdom

RICHARD C. WILSON
Department of Computer Science, University of York
York, YO10 5DD, United Kingdom

EDWIN R. HANCOCK
Department of Computer Science, University of York
York, YO10 5DD, United Kingdom

Received January 3, 2008
Revised October 5, 2008

In this paper, we explore analytically and experimentally a quasi-quantum analogue of the hitting time of the continuous-time quantum walk on a graph. For the classical random walk, the hitting time has been shown to be robust to errors in edge weight structure and to lead to spectral clustering algorithms with improved performance. Our analysis shows that the quasi-quantum analogue of the hitting time of the continuous-time quantum walk can be determined via integrals of the Laplacian spectrum, calculated using Gauss-Laguerre quadrature. We analyse the quantum hitting times with reference to their classical counterpart. Specifically, we explore the graph embeddings that preserve hitting time. Experimentally, we show that the quantum hitting times can be used to emphasise cluster-structure.

Keywords: Continuous time quantum walk, commute time, graph embedding.

Communicated by: R Cleve & J Watrous

1 Introduction

Random walks have been used to formulate a number of powerful algorithms for the analysis of graph structure, and these have found applications in areas such as network analysis, circuit design, pattern recognition and machine learning. One of the attractive features of classical random walks, is that they can be conveniently analysed using graph-spectral methods. For instance, the steady state random walk on a graph is determined by the leading eigenvector of the weighted adjacency matrix or equivalently the Fiedler vector of the Laplacian matrix [17]. In addition, local properties of the random walk such as hitting and commute times can also be conveniently computed using the Laplacian spectrum [7]. Commute time (which for the undirected graphs we are concerned with in this paper is twice the hitting time) has been used to develop graph-spectral clustering algorithms based on embedding that are robust to modifications in local edge structure due to noise [24].

Recently, with the growing interest in quantum computation, quantum walks have been investigated as alternatives to classical walks. The quantum walk differs from the classical walk in that its state vector is complex-valued rather than real-valued and its evolution is described by unitary rather than stochastic matrices. The evolution of the classical walk is random and a state vector can express the probabilities of all the possible outcomes. The evolution of the quantum walk, on the other hand, is deterministic and the randomness only manifests itself when measurements are made. That the state vector of the quantum walk is complex valued allows different paths in the walk to interfere, producing remarkably different probability distributions on the vertices of the graph. Exponential speed-ups have also been observed in the hitting times of particular vertices in some graphs. Szegedy [33] reports similar speed-ups for algorithms based on Markov chains. Magniez *et al* [10] explore the relationships between the hitting times of quantum and random walks on the same structure. For reversible and ergodic Markov chains they report that the quantum analogue of the hitting time goes as the square-root of the classical counterpart.

The aim in this paper is to explore a quasi-quantum analogue of the hitting time of the quantum walk and to investigate whether it offers advantages over the classical commute time for graph embedding. We define the hitting time between a pair of vertices to be the expected time for the walk to arrive at one vertex given that it started at the other *as if we knew the full state vector* of the walk. Note that this would not be possible physically. However, it is analogous to how the hitting time is defined classically and so allows a direct comparison to be carried out.

1.1 Related Literature

1.1.1 Classical Walks

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 [17], and spectral bounds have been placed on the properties of random walks, including the mixing times and hitting times [32].

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 [9]. Of more recent interest is the use of ideas from random walks by Brin and Page to define the Page-Rank index used by the Googlebot search engine [4]. 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 [25, 26] which has used both a standard spectral method [25] and a more sophisticated one based on ideas from graph seriation [26] to convert graphs to strings, so that string matching methods may be used to compare graphs. Gori, Maggini and Sarti [28] on the other hand, have used ideas borrowed from Page-Rank to associate a spectral index with the vertices of a graph and have then used standard attributed graph matching methods to match the resulting attributed graphs.

Melia and Shi [20] have used random walks to learn image segmentation. Zhu, Ghahramani and Lafferty [36] have performed semi-supervised learning using random walks on a labelled graph structure. Borgwardt et al. have developed a kernel that preserves the path length distribution of a random walk on a graph, and have used this to analyse protein data [3]. This kernel has been used by Neuhaus and Bunke [22] to kernelise the computation of graph

edit distance and measure the similarity of graphs. Finally, Qiu and Hancock [24] have shown how the commute times of random walks can be used to render graph-spectral clustering algorithms robust to edge weight errors, and have explored the application of the method to image segmentation, multi-body motion tracking and graph-matching. The commute time allows the vertices of a graph to be embedded in a low dimensional space, and the geometry of this embedding allows the vertices to be clustered into disjoint subsets.

1.1.2 Quantum Walks

Quantum walks have been introduced as quantum counterparts of random walks; a good summary is given by Kempe [14]. As noted above, the behaviour of quantum walks 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.

Quantum walks provide an attractive route to designing quantum algorithms since they lend themselves to physical intuition. There are two different models for the quantum random walk, both of which can be simulated on an arbitrary graph. The first of these is the continuous-time quantum walk proposed by Fahri and Gutmann [11]. They take the vertices of the graph as the basis vectors of the Hilbert space in which the walk takes place. The evolution of the walk is then given by Schrödinger's equation with the graph Laplacian playing the role of a Hamiltonian. The second model is the discrete-time quantum walk, which is the quantum analogue of the classical random walk. The quantum version of the discrete-time quantum walk on an arbitrary graph was formalised by Aharonov, Ambainis and Kempe [1], although the discrete quantum walk had already been considered, albeit in less generality, for example, by Ambainis et al. on the line [2].

Fahri and Gutmann also note that many problems of interest can be formulated in terms of decision trees and solved by considering a random walk on the trees. When a problem is formulated in terms of a decision tree, the question is often to determine whether the tree contains a vertex at the r^{th} level. To solve this problem efficiently it is necessary for the random walk to be able to hit a vertex that is at the r^{th} level in a time polynomial in r. Fahri and Gutmann show that if the classical Markov process is able to penetrate a family of trees in polynomial time, then the quantum walk that they define is also able to. They go on to give an example of a family of trees that cannot be penetrated in polynomial time by the classical Markov process, but can be by their quantum walk. They are not able to use this walk to construct a solution to any classically hard problems and, in fact, show that a different classical algorithm can solve the example that they give. However, their contribution is to show that there may be other ways to achieve algorithmic speed-ups in a manner, which unlike Shor's factorisation algorithm [31], does not rely on the quantum Fourier transform.

By making use of the exponentially faster hitting times that are observed for continuous-time quantum walks on graphs [15] [5], Childs et al showed that it is possible to create an oracle based algorithm using a continuous-time quantum walk that is provably exponentially faster than any possible classical algorithm [6].

An exponential speed-up for the hitting time was observed by Kempe [15], using the discrete-time quantum walk. Kempe considered the walk on the n-dimensional hypercube and was able to show that the hitting time from one vertex to the one opposite is polynomial in n [15]. In general, the discrete-time quantum walk is not easily analysed. Kempe was able to do so for the hypercube due to its regular structure. She demonstrates how this polynomial hitting time could be of use for a routing problem, and also, in a later paper, shows that it could be used to solve the 2-SAT problem efficiently [14] (note that this problem can also be solved efficiently using a classical algorithm). Since it has been demonstrated by Aharonov et al. that the mixing times for discrete-time quantum walk is at most quadratically quicker than that for the classical walk [1], this work demonstrates that hitting times may be of more importance.

Shenvi, Kempe and Whaley [29] showed that the problem of searching an unordered database, previously considered by Grover [12], could be set in the framework of a discrete-time quantum walk. Representing the 2^n elements of a set by the vertices of the n dimensional hypercube, they simulate a walk on the hypercube using a biased coin. The coin is such that its operation on the target state (the element being searched for) is different from its operation on all other states. Again, the walk is started in a superposition of all possible states and, after approximately $\sqrt{N}$ steps, the result of a measurement is the target state with probability $\frac{1}{2} - O(1/n)$. The algorithm is very similar to Grover's algorithm and both make use of the Grover diffusion operator.

1.2 Contribution

The many interesting properties exhibited by quantum walks and the fact that their behaviour is dictated by the structure of the graphs supporting them suggests that they could be a very useful tool for graph analysis. Having said this, there are, as yet, few algorithms based on quantum walks. One approach is that of Shiau, Joynt and Coppersmith [30] who make use of ideas of Rudolph [27] to construct graph invariants based on the walks generated by two particles on a graph obeying Fermi statistics.They then use these invariants to distinguish between graphs.

The aim in this paper is to take the study of continuous time quantum walks one step further. Results for hitting times on certain graphs with high levels of symmetry have been studied [5]. However, to date there has been little effort devoted to the study of properties such as hitting time of walks on more general graphs. Specifically, our aim is to explore whether a quasi-quantum analogue of the hitting time associated with the continuous-time quantum walk can be used to embed the nodes of a graph in a low dimensional vector space and to explore the properties of the embedding.

This is of course not a full quantum analysis of the walk. To do so would require us to specify a measurement process which would lead to the observation of the hitting time. Such a set-up would require a means by which the initial state-vector can be prepared, and then a means by which the state-vector of the walk can be collapsed to yield an observation. This is the approach adopted by Varbanov, Krovi and Brun [19]. Their approach is to measure the quantum walk at random times sampled from a Poisson process. They analyse the dependance of the hitting time of the quantum walk on the rate-parameter of the Poisson process. An alternative is to work with the discrete time quantum walk [21, 35, 16, 15]. However, here the

analysis is not tractable in closed form and the calculation of hitting time must be performed by simulation.

Nonetheless, our analysis leads to a quantity which when contrasted with that for the classical walk possesses a number of interesting properties. First, while the embedding associated with the classical hitting time is close to unidimensional, that associated with the quantum walk needs more dimensions to capture its behaviour. Second, although mean hitting times of the quantum and classical walks are both correlated with path length, in any particular instance they are not correlated with each other: both measure different properties of the graph. Thirdly, the embeddings obtained can distinguish clusters more clearly than those obtained using the classical hitting times.

The outline of this paper is as follows. In Section 2 we review the theory of the classical random walk, its relationship to the Laplacian spectrum and the definition of hitting time. Section 3 introduces the continuous time quantum walk. In Section 4 we show how a hitting time for the continuous time quantum walk can be computed from integrals of the Laplacian spectrum, and how these integrals can be numerically evaluated using Gauss-Laguerre quadrature. Section 5 analyses empirically the properties of the quantum hitting time and compares it with its classical counterpart. In Section 6, we illustrate the application of quantum hitting time to a data clustering problem. Finally, Section 7 offers some conclusions and directions for future research.

2 Graph Laplacian and Classical Hitting Time

In this section, we briefly review some of the properties of the classical random walk on a graph, and the relationship between the hitting time and the Laplacian spectrum. To commence, consider the graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The adjacency matrix of the graph has elements

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

Further, let $D = diag(deg_u; u \in \mathcal{V})$ be the diagonal degree matrix with elements $deg_u = \sum_{v=1}^{n} A(u,v)$, where $n = |\mathcal{V}|$ is the number of vertices in the graph. The Laplacian matrix is given by $L = D - A$. The spectral decomposition of the Laplacian is $L = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is the diagonal matrix with the ordered eigenvalues as elements satisfying the condition $0 = \lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$ and $\Phi = (\phi_1|\phi_2|....|\phi_n)$ is the matrix with the ordered eigenvectors as columns.

Hitting time is a property of a diffusion process on a graph, which is governed by the heat equation, i.e. a partial differential equation associated with the graph Laplacian by

$$\frac{\partial \mathcal{H}_t}{\partial t} = -L\mathcal{H}_t \tag{2}$$

where $\mathcal{H}_t$ is the heat kernel and t is time. The solution of the heat-equation is found by exponentiating the Laplacian spectrum with time i.e.

$$\mathcal{H}_t = \exp[-tL] = \Phi \exp[-t\Lambda]\Phi^T \tag{3}$$

For the nodes u and v of the graph G the element of the matrix is

$$\mathcal{H}_t(u,v) = \sum_{j=1}^{n} \exp[-\lambda_j t]\phi_j(u)\phi_j(v) \tag{4}$$

For a continuous time random walk, the probability state-vector p_t (i.e. the vector whose components are the probabilities of the walk visiting the different nodes of the graph at time t) is also governed by the diffusion equation

$$\frac{\partial p_t}{\partial t} = -Lp_t \tag{5}$$

The solution of the above differential equation is

$$p_t = \exp[-tL]p_0 = \mathcal{H}_t p_0 \tag{6}$$

where p_0 is the initial probability state-vector.

To elucidate the relationship between hitting time and the spectrum of the Laplacian, we need to introduce the pseudo inverse of the Laplacian matrix (or Green's function for the graph)

$$\mathcal{G}(u,v) = \sum_{j=2}^{n} \frac{1}{\lambda_j}\phi_j(u)\phi_j(v) \tag{7}$$

We note that the hitting time $O(u,v)$ of a random walk on a graph is defined as the expected number of steps before node v is visited, commencing from node u. The commute time, $C(u,v)$, on the other hand, is the expected time for the random walk to travel from vertex u to vertex v and then return. For undirected graphs this is simply twice the hitting time. The hitting time $O(u,v)$ is given by [7]

$$O(u,v) = \left\{\mathcal{G}(v,v) - \mathcal{G}(u,v)\right\} \tag{8}$$

Substituting the spectral expression for the Green's function into the definition of the hitting time, it is straightforward to show that in terms of the eigenvectors of the *normalized* Laplacian

$$O(u,v) = \frac{1}{4}\sum_{j=2}^{n} \frac{1}{\lambda_j}\left(\phi_j(u) - \phi_j(v)\right)^2 \tag{9}$$

Let Y be an embedding matrix with the node co-ordinates as columns. The embedding that preserves commute time is

$$Y = \frac{1}{2}\Lambda^{-1/2}\Phi^T \tag{10}$$

The Gram matrix for the embedding is hence

$$Y^T Y = \frac{1}{4}\Phi\Lambda^{-1}\Phi^T = \frac{1}{4}\mathcal{G} \tag{11}$$

So the Green's function (pseudo inverse) is a multiple of the kernel matrix for the embedding.

3 The Continuous-time Quantum Walk

Like the classical random walk on a graph, the state-space of the continuous-time quantum walk is the set of vertices of the graph. However, the probability of being at a certain state is given by the square of the amplitude of that state, rather then just the amplitude of the state (as is the case classically). This allows destructive as well as constructive interference to take place.

The state space for the continuous-time quantum walk on a graph, $G = (\mathcal{V}, \mathcal{E})$, is the set of vertices, $\mathcal{V}$, as is the case for the classical random walk. In addition, transitions only occur between adjacent vertices. If the walk is at vertex u, the components of the state vector corresponding to vertices adjacent to u change at a rate proportional to $\frac{1}{d(u)}$. The basis states for the continuous-time quantum walk are vectors corresponding to particular vertices, as is the case for the classical random walk. The basis state corresponding to the walk being at $u \in \mathcal{V}$ is written, in Dirac notation, as $|u\rangle$. A general state of the walk is a complex-linear combination of these basis states and so the state of the walk at time t is given by a vector, $|\psi_t\rangle \in \mathbb{C}^{|\mathcal{V}|}$, which we write componentwise as

$$|\psi_t\rangle = \sum_{u \in \mathcal{V}} \alpha_u(t)|u\rangle. \tag{12}$$

Thus, the amplitudes $\alpha_u(t) \in \mathbb{C}$.

If measured, the probability of the walk being in a particular state is given by the square of the amplitude of that state. In practice, the measurement process alters the state. As a result the walk is not in state $|\psi_t >$ but is projected onto $|u >$. So measuring the walk alters its state and hence its subsequent dynamics. Hence, our definition of hitting time is not accurate from the perspective of quantum observables, but instead a classical average. Nonetheless, let X_t be the random variable giving the location of the walk at time t. The probability of being at $u \in \mathcal{V}$ at time t is given by

$$\Pr(X^t = u) = \alpha_u \alpha_u^* \tag{13}$$

where α_u^* is the complex conjugate of α_u. As the total probability must sum to unity, we have that $|\alpha_u(t)| \in [0, 1]$ for all $u \in \mathcal{V}$, $t \in \mathbb{R}^+$, and $\sum_{u \in \mathcal{V}} \alpha_u(t)\alpha_u^*(t) = 1$ for all $t \in \mathbb{R}^+$.

The evolution of the state vector is given by

$$\frac{d}{dt}|\psi_t\rangle = -iL|\psi_t\rangle. \tag{14}$$

We note that, unlike the classical walk, the quantum walk is not strictly a Markov chain. That is, the evolution of the probability vector of the walk being measured at time t depends on the state vector of the walk, and not the probability vector alone. However, the state vector itself is a Markov chain.

Given an initial state for the walk, $|\psi_0\rangle$, Equation 14 can be solved to give

$$|\psi_t\rangle = e^{-iLt}|\psi_0\rangle. \tag{15}$$

Thus, given the initial state, we can calculate the state of the walk at an time, t.

4 The Quantum Hitting Time

In this section we develop a measure of the hitting time for the continuous-time quantum walk based on having full knowledge of the state vector of the walk. Mathematically, this is clearly possible. However, since we do not base this concept on a realisable measurement process, the hitting time developed does not correspond to a physically observable quantity. Instead, we compute a quasi-quantum analogue of the hitting time for the continuous time quantum walk. This commences by recalling that the classical hitting time is the expected value of the time taken for the walk to traverse the set of paths between a pair of nodes. We make use of the state vector of the continuous-time quantum walk to compute the transition probabilities required in this computation. An experimental comparison of the resulting hitting time reveals deep differences between the classical and quantum continuous time walks.

To commence, we consider the expected time for the quantum walk to travel between each pair of vertices in a graph. For the walk with starting state $|\psi_0\rangle = |u\rangle$, let $X^t_{(u)}$ be the observed random variable giving the measured state of the walk at time t based on full knowledge of the state vector. That is, we define $X^t_{(u)}$ to be such that $\Pr(X^t_{(u)} = v) = |\langle v|e^{-iLt}|u\rangle|^2$. Similarly, let $H(u,v)$ be the random variable giving the first hitting time of the vertex v. That is, $H(u,v) = \min\{t|X^t_{(u)} = v\}$. We note that, due to the symmetry of the Laplacian matrix, L, the first hitting times are symmetric, that is, $H(u,v) = H(v,u)$. The expected hitting time between a pair of vertices, $O(u,v)$, can be calculated using

$$O(u,v) = \int_0^\infty \Pr\big(H(u,v) = t\big)t\,dt \tag{16}$$

Of course, the above definition violates the strict notion of observability in quantum systems. While $X^t_{(u)}$ has a physical meaning, $H(u,v)$ has none since it would require the walk to be continually measured. This of course would alter the dynamics of the walk.

To calculate $\Pr\big(H(u,v) = t\big)$, we consider the state vector, $|\psi_t\rangle = e^{-iLt}|u\rangle$. Let the number of vertices in the graph be n and let the eigenvalues and corresponding eigenvectors of L be λ_k and ϕ_k respectively ($1 \leq k \leq n$). The probability of observing the walk at node v at time t is given by

$$\begin{aligned}
\Pr(X^t_{(u)} = v) &= |\langle v|e^{-iLt}|u\rangle|^2 &(17)\\
&= |\langle v|\Phi e^{-i\Lambda t}\Phi^T|u\rangle|^2 &(18)\\
&= \left(\sum_{j=1}^{n}\phi_j(u)\phi_j(v)e^{-i\lambda_j t}\right)\left(\sum_{k=1}^{n}\phi_k(u)\phi_k(v)e^{-i\lambda_k t}\right)^* &(19)\\
&= \sum_{j=1}^{n}\sum_{k=1}^{n}\phi_j(u)\phi_j(v)\phi_k(u)\phi_k(v)e^{-i(\lambda_j-\lambda_k)t}. &(20)
\end{aligned}$$

As $P(X^t_{(u)} = v)$ is a real number we need only consider the real parts, since the imaginary parts must cancel. Hence, we have that

$$\Pr(X^t_{(u)} = v) = \sum_{j=1}^{n} \left(2 \sum_{k>j:\ \lambda_j \neq \lambda_k} \phi_j(u)\phi_j(v)\phi_k(u)\phi_k(v)\cos((\lambda_k - \lambda_j)t) \right. \tag{21}$$

$$\left. \cdots + \sum_{k:\ \lambda_j = \lambda_k} \phi_k^2(u)\phi_k^2(v) \right) \tag{22}$$

$$= F + \sum_{(A,B)\in\mathcal{X}} A\cos(Bt), \tag{23}$$

where the pairs $(A, B) \in \mathcal{X}$, and are defined by

$$\mathcal{X} = \left\{ \left(\phi_j(u)\phi_j(v)\phi_k(u)\phi_k(v), \lambda_j - \lambda_k \right) \middle| 1 \leq j \leq n,\ 1 \leq k < j,\ \lambda_k \neq \lambda_j \right\} \tag{24}$$

and

$$F = \sum_{j,k:\ \lambda_j = \lambda_k} \phi_j(u)\phi_j(v)\phi_k(u)\phi_k(v). \tag{25}$$

For a particular pair of vertices $u, v \in \mathcal{V}$, let $r(t)$ be the probability density function for the first hitting time, $r(t) = \Pr(H(u, v) = t)$, and $R(t) = \int_0^t r(t')dt'$ the cumulative distribution function. The initial conditions for $R(t)$ are $R(0) = 0$ if $u \neq v$ and $R(0) = 1$ if $u = v$. The quantity $P(X^t_{(u)} = v)$ can be regarded as the hazard function for the walk. It is such that the probability of the walk arriving at v in the time interval $(\tau, \tau + d\tau)$, given that it has not previously arrived at v, is given by $P(X^\tau_{(u)} = v)d\tau$. Here we diverge from the approach taken by Varbanov, Krovi and Brun [19]. We take a classical viewpoint and assume that the arrival of the walk can be observed in the time interval. Varbanov, Krovi and Brun on the other hand, adopt a quantum observation process where the observation times are sampled from a Poisson distribution.

Thus, for a given hazard function $P(X^t_{(u)} = v)$, we have that

$$\frac{d}{dt}(1 - R(t)) = -P(X^t_{(u)} = v)(1 - R(t)). \tag{26}$$

Hence, integrating both sides with respect to time we have

$$1 - R(t) = (1 - R(0)) \exp\left(- \int_0^t F + \sum_{(A,B)\in\mathcal{X}} A\cos(B\tau)\, d\tau \right) \tag{27}$$

$$= \exp\left(- Ft - \sum_{(A,B)\in\mathcal{X}} \frac{A}{B}\sin(Bt) \right), \tag{28}$$

if $u \neq v$ and $R(t) = 1$ if $u = v$. Thus, $O(u, u) = 0$ for all $u \in \mathcal{V}$.If $u \neq v$, the probability density function for the random variable $H(u, v)$ is then given by

$$r(t) = (1 - R(t)) \Pr(X^t_{(u)} = v). \tag{29}$$

Hence, by substitution

$$r(t) = \left\{ \exp\left(- Ft - \sum_{(A,B)\in\mathcal{X}} \frac{A}{B}\sin(Bt) \right) \right\} \left(F + \sum_{(A,B)\in\mathcal{X}} A\cos(Bt) \right). \tag{30}$$

Thus, using Equation 16, we can calculate the hitting time for the vertex v for the walk starting at u.

$$O(u,v) = \int_0^\infty t\Big(F + \sum_{(A,B)\in\mathcal{X}} A\cos(Bt)\Big)e^{-Ft-\sum_{(A,B)\in\mathcal{X}}\frac{A}{B}\sin(Bt)}dt \tag{31}$$

$$= \Big[-te^{-Ft-\sum_{(A,B)\in\mathcal{X}}\frac{A}{B}\sin(Bt)}\Big]_0^\infty + \int_0^\infty e^{-Ft-\sum_{(A,B)\in\mathcal{X}}\frac{A}{B}\sin(Bt)}dt \tag{32}$$

$$= \int_0^\infty e^{-Ft-\sum_{(A,B)\in\mathcal{X}}\frac{A}{B}\sin(Bt)}dt \tag{33}$$

In order to calculate this numerically we use Gauss-Laguerre quadrature [13]. Given a function, $f \in L^1(0,\infty)$ we have that

$$\int_0^\infty f(x)dx = \int_0^\infty e^{-x}[e^x f(x)]dx \simeq \sum_{k=1}^n w(x_k)e^{x_k}f(x_k). \tag{34}$$

The abscissas, for x_k $(k = 1,\ldots,n)$ for the quadrature order n are given by the roots of the n^{th} order Laguerre polynomial

$$L_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}(e^{-x}x^n) \tag{35}$$

and the weights are

$$w(x_k) = \frac{x_k}{(n+1)^2(L_{n+1}(x_k))^2}. \tag{36}$$

In the remainder of the this paper, we refer to the quasi-quantum analogue of the hitting time of the continuous time quantum walk simply the "quantum hitting time".

At this point it is worth pausing to consider the relationship between our work and that of Varbanov, Krovi and Brun [19], who base their observation process on random samples from a Poisson distribution. Their estimate of the hitting time therefore depends on the rate parameter of the Poisson process, and can mis-estimate the hitting time if it is mismatched to the rate parameter. As noted earlier, we do not specify a measurement process, but instead use $\Pr(X^t_{(u)} = v)$ to compute a density function $r(t)$ for the classical hitting time. We could modify our definition to include a Poisson sampling process of the type used by Varbanov, Krovi and Brun. However, the resulting estmates would be dependant on the rate-parameter and would be of limited use in the experimental study conducted later. Our approach is hence only pseudo-quantum.

5 Analysis of the Quantum Hitting Time

In this section we analyse the differences between the quantum and classical hitting times and their corresponding embeddings. We begin by considering the hitting times on graphs randomly generated according to one of two different models. The first model is the randomly connected graph. Here edges between the n vertices are randomly generated with connection probability p. Our second model is the banded adjacency graph [8]. The banded adjacency graphs have three parameters: n is the number of vertices, b is the width of the band in which vertices can be connected, and p is the probability that a pair of vertices in this band are

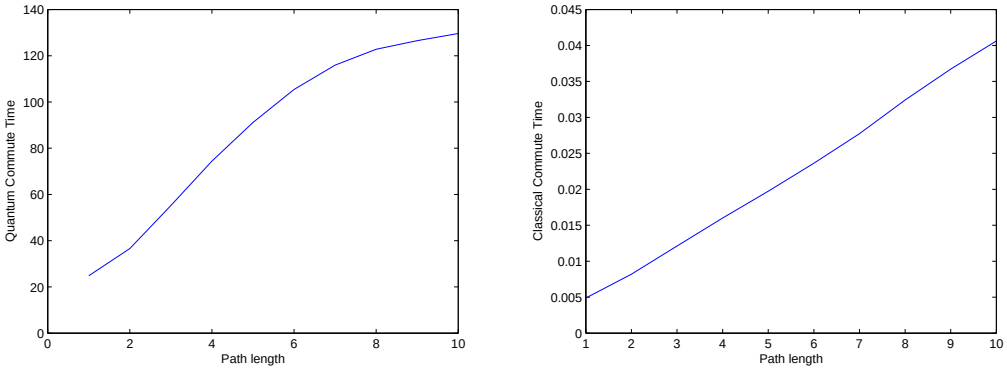

Fig. 1. The average quantum hitting time (left) and average classical hitting time (right) as a function of path length for a banded adjacency graph on 50 vertices with band width 10 and p=0.3.

connected. For the purposes of constructing each graph we take a set of vertices numbered 1 to n. With probability p, the vertex u is connected to each vertex, $v \in \{u-b, u-b+1, \ldots, u+b\}$. In terms of its adjacency matrix, such a graph only has non-zero entries within a band of width $2b$ centred on the main diagonal. Such graphs approximate many real-world structures, for example, molecules or VLSI circuits. The randomly generated graphs model graphs where no such restriction is obvious.

5.1 Hitting Times and Path Length

We begin by considering the relationship between hitting times and path length. The path length between a pair of vertices is the number of edges that must be traversed along the shortest path between them. It is a less sophisticated measure of distance than hitting time since it only takes into account the single shortest path between a pair of vertices. The hitting time, on the other hand, decreases as the number of alternative paths between the vertices increases. This expresses mathematically the understanding that a pair of vertices with many paths between them of length l should be considered more closely connected than a pair of vertices with just one path of length l between them.

Figure 1 shows the mean quantum hitting times and mean classical hitting times as a function of path length. We see that the functions are both monotonically increasing. However, the quantum hitting time has a non-linear relationship with path length. Larger path lengths do not correspond to proportionally larger quantum hitting times, as they do in the classical case. Similar behaviour has been observed for the quantum walk on the line [21] and the circle [14] where hitting times were shown to be quadratically faster. We will see that this behaviour has important consequences when we consider the treatment of outliers in the embedding of graphs via hitting times.

Although Figure 1 shows that the mean quantum and classical hitting times are functions of path length, the standard deviation about these functions are large. Thus, we wish to investigate if there is a relationship between the quantum hitting time, $Q(u, v)$, and the classical hitting time, $C(u, v)$, for a given pair of vertices. Figure 2a shows a scatter plot

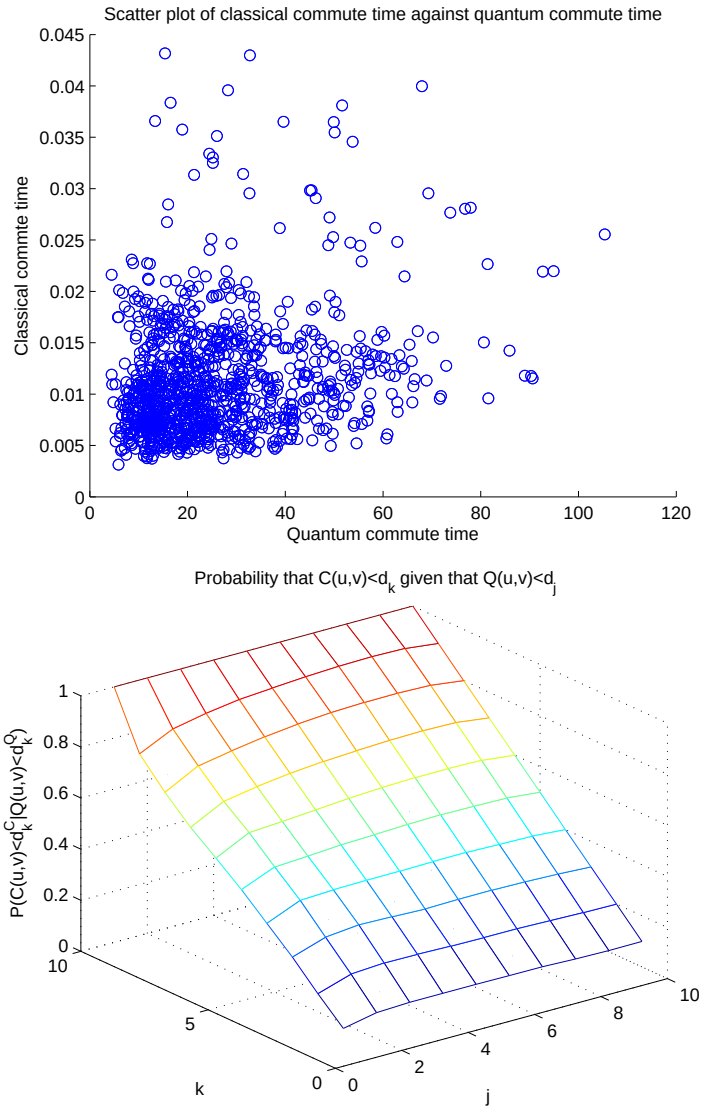

Fig. 2. Relationship between Classical and Quantum Hitting Times: a) Top: Scatter plot of classical hitting times against the quantum hitting times for 10 graphs. b) Bottom: The probability that the classical hitting time between a pair of vertices is less than the k^{th} decile of the classical hitting times between all pairs of vertices in the graph, given that the quantum hitting time is less than the j^th decile of the quantum hitting times. We use 10 graphs with $n = 35$, $p = 0.3$ for both plots.

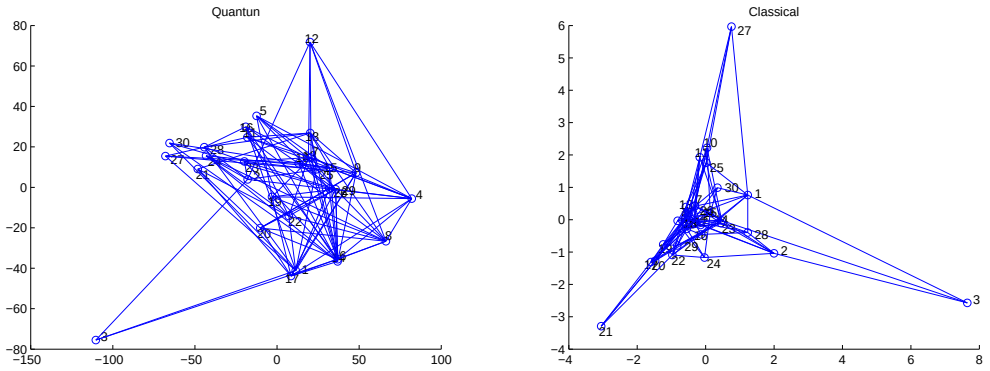

Fig. 3. Quantum Hitting Time Embeddings of Random Graphs: Graph embeddings using the quantum hitting time (left) and the classical hitting time (right) for a graph with 30 vertices and probability of each pair of vertices being connected 0.3

of the classical hitting times between pairs of vertices versus the quantum hitting times for a set of 10 randomly connected graphs with $n = 10$, $p = 0.3$. We see that there is very little correlation between the hitting times for particular vertices. However, as the classical (or quantum) hitting time increases, the lower bound for the corresponding quantum (or classical) hitting time does also increase by a proportionally smaller amount. This shows that given a classical/quantum hitting time, the corresponding quantum/classical hitting time cannot be arbitrarily small.

To further investigate any correlation, let d_k^C be the k^{th} decile for the classical hitting times and d_j^Q the j^{th} decile for the quantum hitting times for a particular graph. We consider the set of probabilities $\Pr(C(u,v) < d_k^C | Q(u,v) < d_j^Q)$. Figure 2b shows a plot of this function. The figure shows that $\Pr(C(u,v) < d_k^C)$ is almost completely independent of any condition of the form $Q(u,v) < d_j^Q$. This demonstrates that the two hitting times emphasise different measures of distance.

5.2 Quantum and Classical Hitting Time Embeddings

We now give some example embeddings, using both the classical and quantum hitting times. In the case of the classical hitting time the matrix of embedding co-ordinates is given by $Y = \frac{1}{2}\Lambda^{-1/2}\Phi^T$. The vectors of embedding co-ordinates for the nodes of a graph are the columns of the matrix Y. We confine or attention to embeddings on the plane, and so we use just the first two rows of Y to give us the 2D embedding co-ordinates.

The quantum hitting time, on the other hand, is evaluated numerically. To perform the embedding, we therefore use multidimensional scaling (MDS) [34]. 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 nodes in the graph under consideration. The first step of MDS is to calculate a similarity matrix, F, whose element with row r and column c is given by $F(r,c) = -\frac{1}{2}\{C(r,c) - \hat{C}(r,.) - \hat{C}(.,c) + \hat{C}(.,.)\}$, where $\hat{C}(r,.) = \frac{1}{n}\sum_{c=1}^{n} C(r,c)$ is the average hitting-time for the r^{th} row, $\hat{C}(.,c)$ is the average hitting-time over the c^{th} column and $\hat{C}(.,.) = \frac{1}{n^2}\sum_{r=1}^{n}\sum_{c=1}^{n} C(r,c)$ is the average hitting-time over all rows and columns.

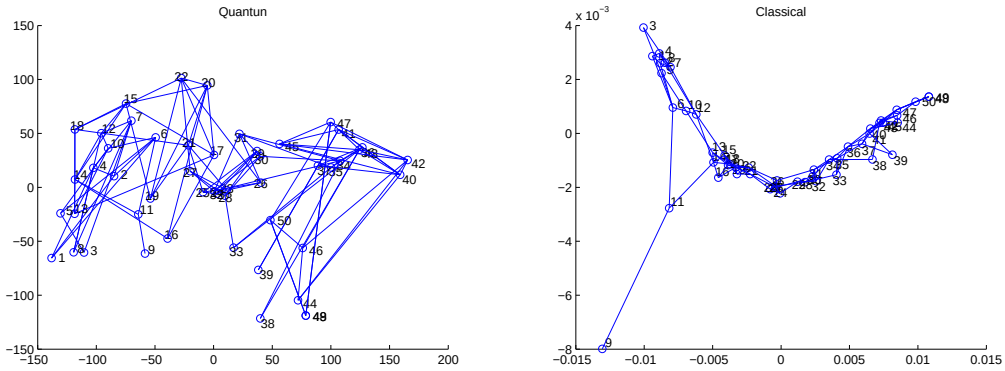

Fig. 4. Quantum Hitting Time Embeddings of Banded Adjacency Graphs: Graph embeddings using the quantum hitting time (left) and the classical hitting time (right) for a banded adjacency graph with 50 vertices, band width $b = 5$ and probability of each pair of vertices in the band being connected $p = 0.5$

We perform the eigendecomposition $F = \Psi E \Psi^T$ on the disimilarity matrix, where E is the diagonal matrix with the ordered eigenvalues along the diagonal, and Ψ is the matrix with the correspondingly ordered eigenvectors as columns. If the rank of F is k, where $k < n$, then we will have k non-zero eigenvalues. The matrix with the vectors of embedding coordinates for the nodes as columns is $Y = \sqrt{E}\Psi^T$. We again take the first two rows of Y to give us a 2D visualisation of the graph.

Figure 3 shows an example of the quantum and classical hitting time embeddings for a randomly connected graph. The most obvious difference between the two embeddings is the classical hitting time embedding's tendency to produce a few outliers while confining the majority of the vertices to a small area. The problem of one distance outweighing all remaining ones is much less of a problem for the quantum hitting time embedding, and consequently the vertices are distributed more evenly.

Embeddings of a banded adjacency graph are given in Figure 4. The classical hitting time embedding allows the distance along the band to outweigh the remaining distances, causing the graph to lie almost entirely along a curve, obscuring almost all of the local structure. The quantum hitting time embedding, however, shows both the band structure (principally captured by the x-component of the embedding) and the local structure along the band (principally captured by the y-component).

5.3 Symmetric Graphs

We now turn our attention to graphs which possess some form of symmetry. If the graph on which the walk takes place has some kind of symmetry, certain paths can cancel each other out while others reinforce each other. It is also on graphs with symmetries that exponentially faster hitting times have been observed and so we would expect the quantum and classical embeddings for such graphs to be different. Two graphs on which this has been observed are the hypercube [15] and the graph obtained by randomly connecting a pair of n-level binary trees at the nodes [6].

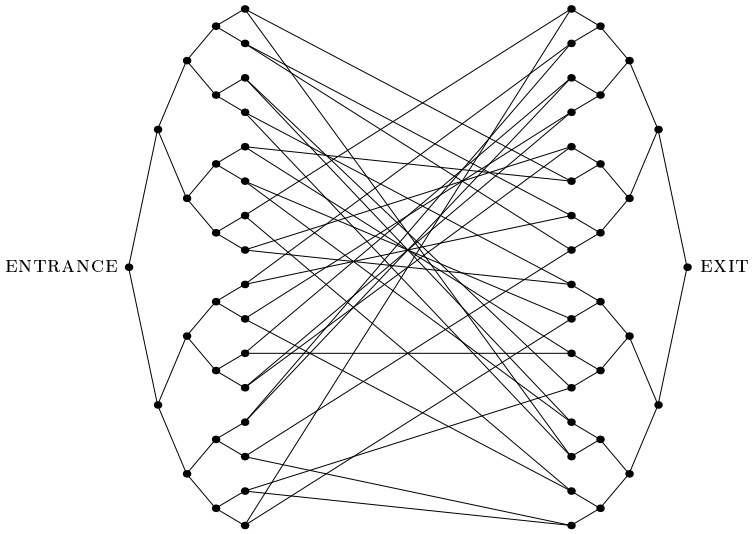

Fig. 5. Graph formed by randomly connecting a pair of 5-level binary trees at the 'leaves'.

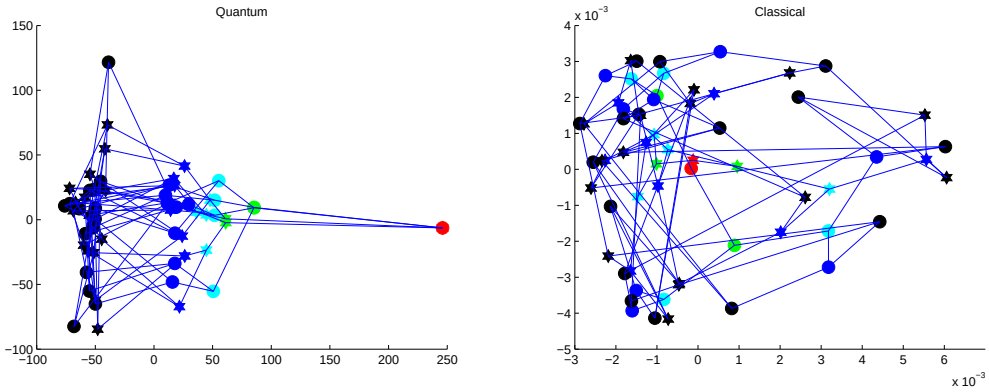

Fig. 6. Hitting Time Embeddings of the Conjoined Binary Tree: Quantum and classical hitting time embeddings of the pair of 5-level binary trees joined at the the 'leaves'. The structure of the graph can be seen in Figure 5. The vertices of the one graph are plotted as circles, those of the other as stars. Vertices at different levels of the trees are different colours.

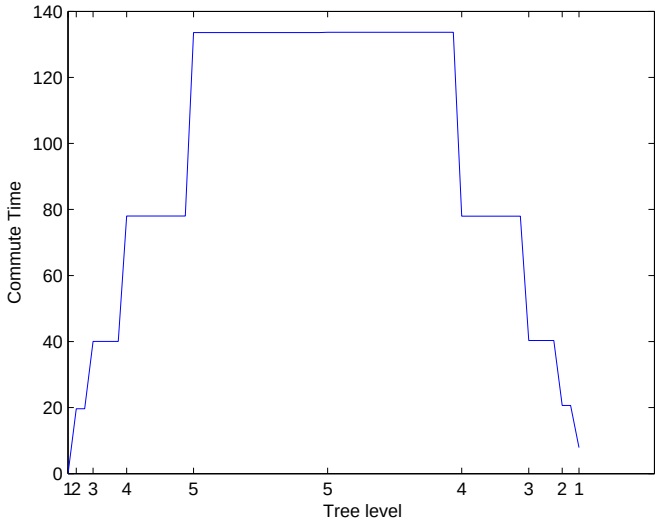

Fig. 7. Quantum Hitting Times on the Conjoined Binary Tree: The quantum hitting time from one root vertex to all the other vertices in the graph.

In Figure 5 we show a graph formed by randomly connecting a pair of 5-level binary trees at the 'leaves'. Figure 6 gives the classical and quantum hitting time embeddings for this graph. We can see that whereas the classical hitting time embedding does not reveal the structure of the graph, the quantum hitting time embedding does. Furthermore, the level of a vertex in one of the two trees is given by the first component of the embedding, and different vertices at each level are separated by the second component of the embedding. However, the graph appears as though it has been folded in half. For example, the hitting times from one of the root vertices to all other vertices in the graph is very similar to the hitting times to those vertices from the other root vertex. This is regardless in which of the two original trees the vertices lie. Consequently, the two root vertices are almost exactly coincident in the quantum hitting time embedding (the red circle and red star in Figure 6). Figure 7 shows the hitting time from one of the root vertices to all the other vertices. This figure shows that the hitting time from the root vertex to a vertex in the n^{th} level of the same tree as itself is almost exactly the same as the hitting time to a vertex in the n^{th} level of the other tree.

Figure 8 shows the embeddings of the 3 by 3 grid. The classical hitting time embedding of the graph clearly shows the structure of the graph. However, as we saw for the joined binary trees, a number of vertices in the quantum hitting time embedding are coincident. For the grid, the corner vertices are coincident with each other, the opposite sides of the grid are each coincident, and the central vertex is plotted singularly. This again reveals the symmetries in the quantum hitting times.

We can consider the pattern of quantum hitting times on a graph in terms of which dimension of an MDS embedding would separate which vertices. Figure 9 shows this for the 3 by 3 grid. We see that different dimensions of the embedding identify different symmetries of the graph by plotting different vertices as coincident points. Although opposite corners of the grid are coincident in the 2D embedding, the hitting times from these vertices to each

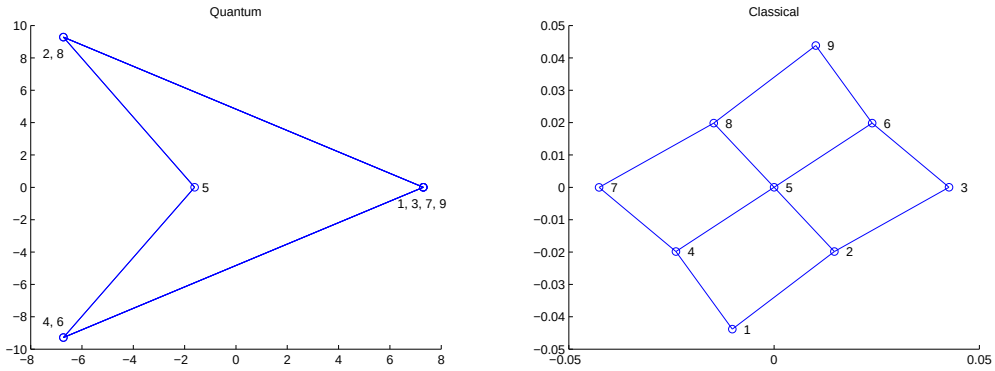

Fig. 8. Quantum and classical hitting time embeddings of the 3 by 3 grid with the vertices labelled.

of the other vertices in the grid are not identical, and consequently the 5^{th} dimension of the embedding separates them. Figure 10 shows the symmetries of the 8×8 grid identified by the quantum hitting time embedding. As before, various symmetries between opposite vertices are identified.

We now consider the graph obtained by making connections between the vertices in a cycle. We connect 2 pairs of vertices from the cycle with 32 vertices to give a graph we refer to as a 'wheel' with 4 'spokes'. The hitting time embeddings are shown in Figure 11. In the two dimensional quantum embedding the vertices connected to the spokes are all coincident and a four-fold symmetry has been identified. The symmetries of the first four dimensions of the embedding are shown in Figure 12. The third dimension of the embedding, for example, identifies a different four-fold symmetry.

6 Experiments

We consider the problem of emphasising the cluster structure of a dataset. We do this using distance between the Laplacian spectra of a set of graphs. Given a set of graphs we calculate distances between them using the Euclidean distance between their Laplacian spectra. We use the Laplacian spectra as a vehicle to test our approach, its use in this context has been well studied in the literature [18]. We represent the set of graphs using a weighted data similarity graph (WDSG). The vertices of the WDSG graph are the original graphs from the set we wish to cluster. We then use the classical and quantum hitting times to embed these graphs. We also embed the graphs directly using MDS directly on the matrix of distances for comparison.

Let $\mathcal{S}$ be our set of m graphs. We denote by $\mathbf{z}_k$ the vector of the ordered eigenvalues of the Laplacian matrix, L_k, of the the k^{th} graph. We append zeros to the shorter vectors so that all the vectors of eigenvalues are of the same length. The WDSG representing this dataset is the complete graph $G = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{1, \ldots, m\}$, $\mathcal{E} = \{\{u, v\} | u, v \in \mathcal{V},\ u \neq v\}$ and the weights are given by $W(u, v) = e^{-||\mathbf{z}_u - \mathbf{z}_v||}$, where $||\mathbf{z}_u - \mathbf{z}_v||$ is the standard Euclidean distance.

We use the standard CMU, MOVI and chalet house sequences as our dataset [18]. The

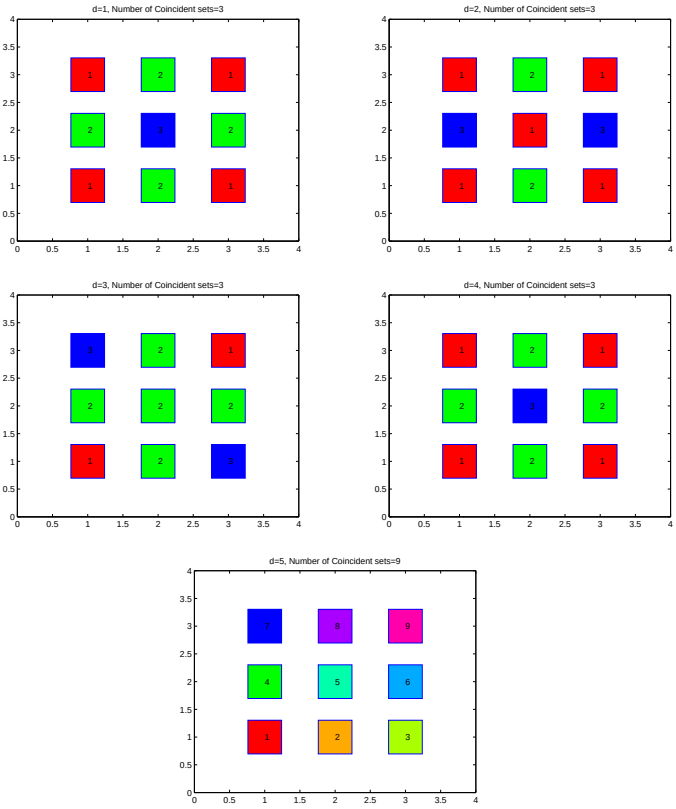

Fig. 9. Coincident vertices in the d^{th} dimension of the quantum hitting time embedding of the 3×3 grid. Coincident vertices are labelled with the same colour and number.

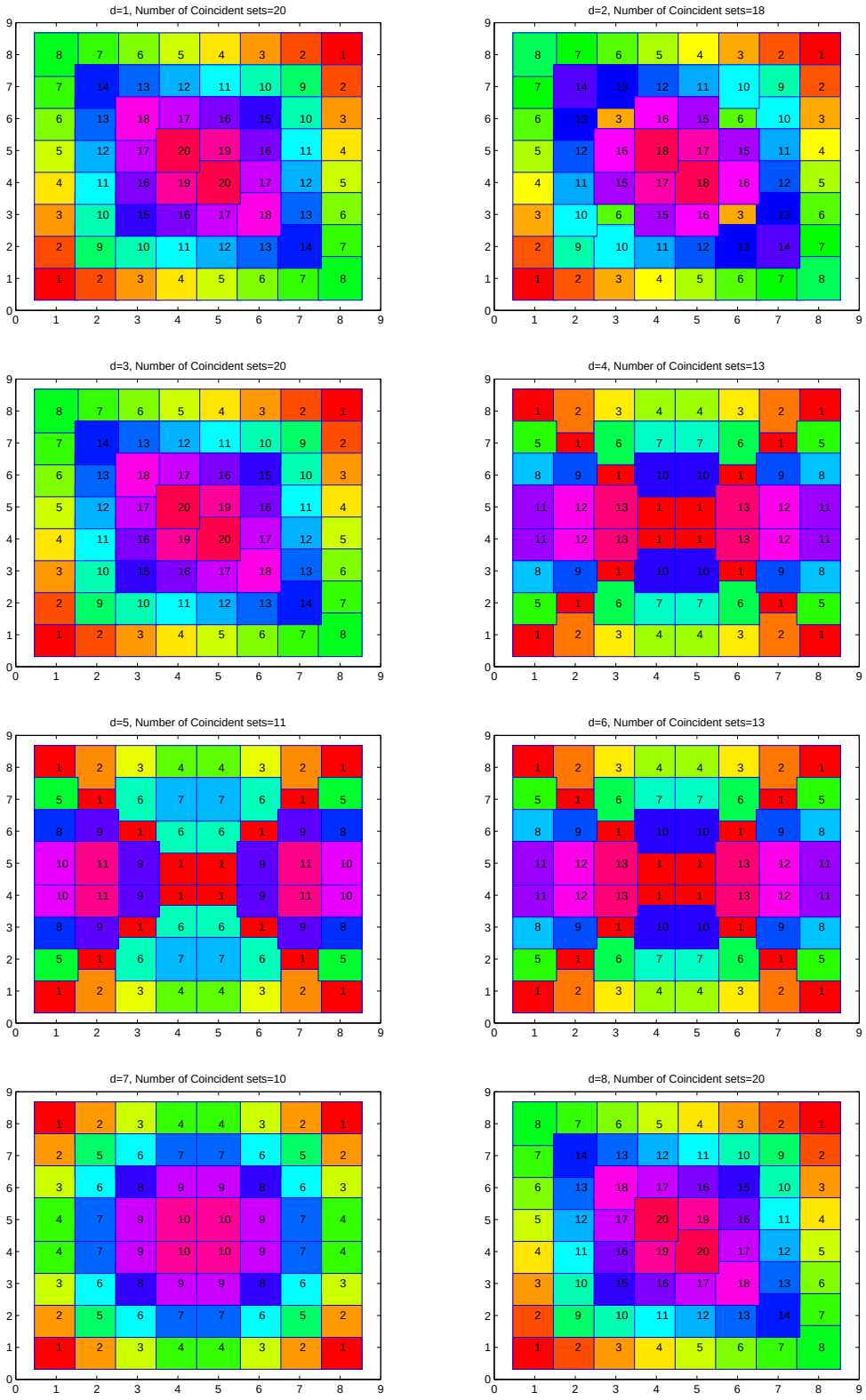

Fig. 10. Coincident vertices in the d^{th} dimension of the quantum hitting time embedding of the 8×8 grid. Coincident vertices are labelled with the same colour and number.

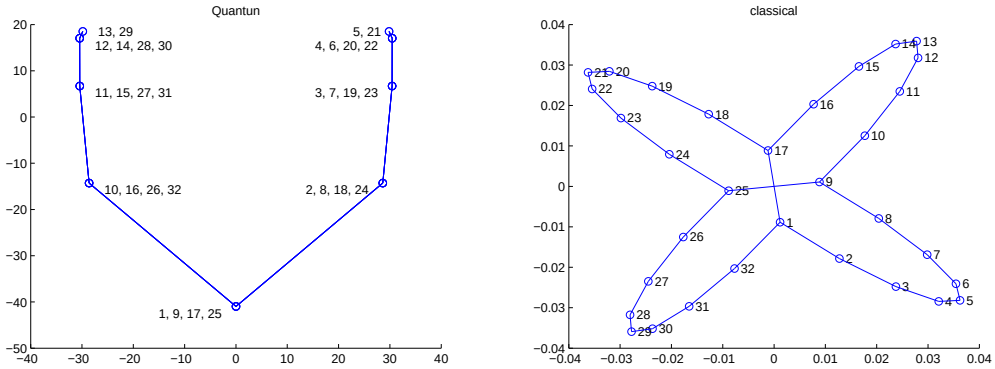

Fig. 11. Quantum and classical hitting time embeddings of the 32 vertex 'wheel' with 4 'spokes'.

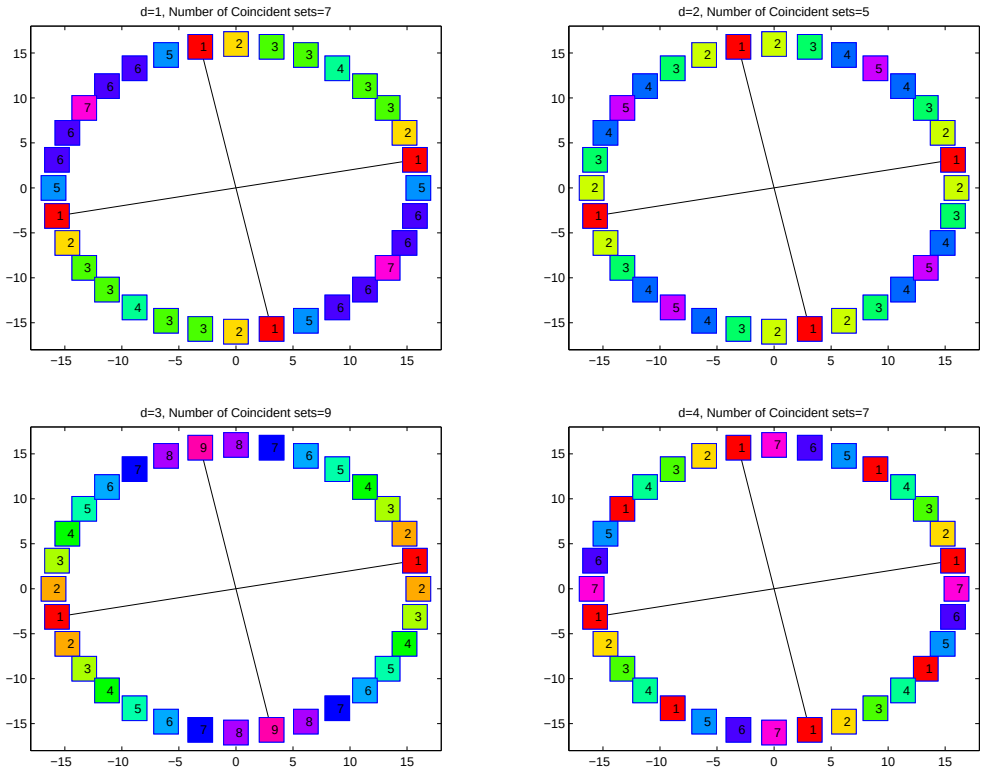

Fig. 12. Coincident vertices in the d^{th} dimension of the quantum hitting time embedding of the 32 vertex 'wheel' with 4 'spokes'. Coincident vertices are labelled with the same colour and number.

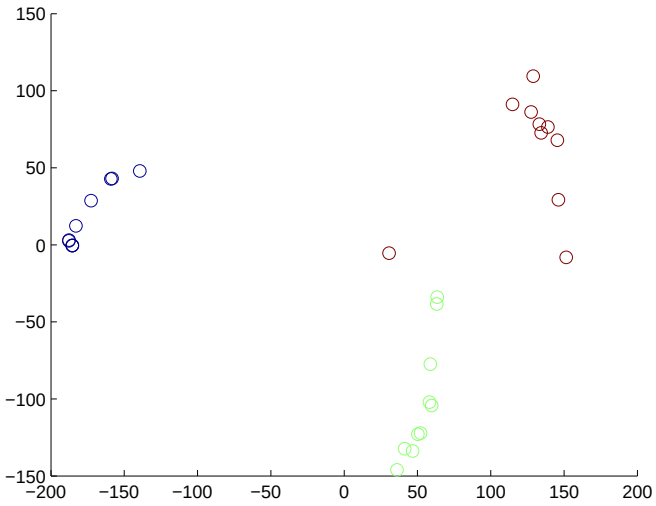

Fig. 13. Embeddings of the house sequences using the spectral and the quantum hitting time.

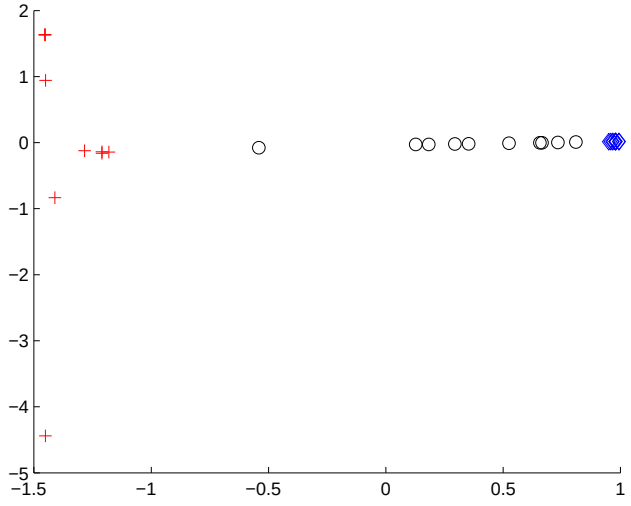

Fig. 14. Embeddings of the house sequences using MDS on the classical hitting times.

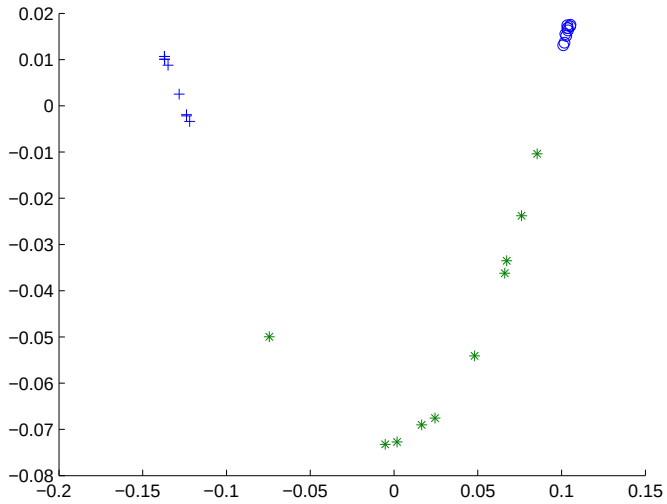

Fig. 15. Embeddings of the house sequences using MDS on the spectral distances directly.

houses are viewed at angular intervals and Delaunay triangulations are derived from these to form the set of graphs. The embedding obtained using the quantum hitting time is shown in Figure 13, the embedding using the classical hitting time is shown in Figure 14 and the embedding using MDS directly on the matrix of spectral distances is shown in Figure 15. The embedding using the quantum hitting times clearly distinguishes between the 3 clusters. A reasonably good separation of the houses is also obtained using MDS on the spectral distances directly. However, the embedding using MDS on the spectral distances directly is effectively one dimensional, this is a common problem with using MDS on graph spectral distances [18]. For datasets with more than three classes this is likely to prevent this method from successfully clustering different classes separately. The quantum hitting time embedding, however, fully utilizes the available dimensions.

The embedding using the quantum hitting times distinguishes between the different classes far more clearly than the embedding using the classical hitting time is able to. The embedding using the classical hitting time produces a number of outliers resulting in the majority of the nodes being confined to a very small area of the embedding. What is more, looking at this central cluster where the majority of the nodes lie shows that, even here, the nodes are not clearly clustered according to their separate classes.

7 Conclusions

We have shown how a quasi-quantum analogue of the hitting time between a pair of vertices in a graph can be calculated for the continuous time quantum walk on a graph. The hitting time can be used as a more robust measure of distance between vertices than the path length between vertices. We analyse how the quantum hitting time compares with the classical hitting time which was considered in [23]. The hitting times between the vertices of a graph can be used to embed that graph in a low dimensional space. We show how the quantum

hitting time produces embeddings of graphs that are prone problems caused by outliers than classical hitting time embeddings. In addition, quantum hitting time embeddings make full use of both dimensions of the 2D embedding space rather than restricting the graph to a submanifold, as can occur with classical hitting time embeddings. We show how quantum hitting time embeddings can be used to cluster real-world datasets effectively. Additionally, we show how the quantum hitting time embedding can be used to identify symmetries in graphs.

There are number of directions for future work. First, as mentioned in the introduction, our analysis falls well short of defining a physical observable of the continuous-time quantum walk since we do not specify a measurement process. This would require a deeper analysis of the dynamics of the walk, and it is not clear that a convenient observable or a corresponding measurement process exists. Second, it would be interesting to extend the experimental analysis provided in the paper to hitting times computed for the discrete-time quantum walk .

References

1. D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *STOC '01: Proc ACM Theory of computing*, pages 50–59. ACM Press, 2001.
2. A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous. One-dimensional quantum walks. In *Proc. 33th STOC*, pages 60–69, New York, NY, 2001. ACM.
3. K. Borgwardt, S. Schnauer, S. Vishwanathan, A. Smola, and H. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:147–156, June 2005.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
5. A. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1:35, 2002.
6. A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. In *STOC '03: Proc. 35th ACM symposium on TOC*, pages 59–68, New York, NY, USA, 2003. ACM Press.
7. F. Chung and S.-T. Yau. Discrete green's functions. *J. Comb. Theory Ser. A*, 91(1-2):191–214, 2000.
8. F. DePiero and J. Carlin. Structural matching via optimal basis graphs. In *ICPR '06*, pages 449–452, 2006.
9. P. Doyle and J. Snell. *Random Walks and Electrical Networks.* Mathematical Association of America, Washington, 1984.
10. P.C. Richter F. Magniez, A. Nayak and M. Santha. On the hitting times of quantum versus random walks. *arXiv:0808.0084v1 [quant-ph]*, 2008.
11. E. Farhi and S. Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58:915–928, 1998.
12. L. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, NY, 1996. ACM Press, New York.
13. F. B. Hildebrand. *Introduction to numerical analysis.* McGraw-Hill, New York, 2^{nd} edition, 1973.
14. J. Kempe. Quantum random walks—an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.
15. J. Kempe. Quantum random walks hit exponentially faster. In *Proc. of 7th Intern. Workshop on Randomization and Approximation Techniques in Comp. Sc. (RANDOM'03)*, pages 354–69, 2003.
16. H. Krovi and T.A. Brun. Hitting time for quantum walks on a hypercube. *Phys. Rev. A*, 73:032341,

2006.
17. L. Lovász. *Combinatorics, Paul Erdös is Eighty*, volume 2, chapter Random Walks on Graphs: A Survey, pages 353–398. János Bolyai Mathematical Society, Budapest, 1996.
18. B. Luo, R. C. Wilson, and E. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213–2230, 2003.
19. H. Krovi M. Varbanov and T.A. Brun. Hitting time for the continuous time quantum walks. *arXiv:0803.3446v1 [quant-ph]*, 2008.
20. M. Meila and J. Shi. A random walks view of spectral segmentation. In *AI and STATISTICS (AISTATS) 2001*, 2001.
21. A. Nayak and A. Vishwanath. Quantum walk on the line. *arXiv:quant-ph/0010117 [quant-ph]*, 2000.
22. M. Neuhaus and H. Bunke. A random walk kernel derived from graph edit distance. In *SSPR*, pages 191–199, 2006.
23. H. Qiu and E. Hancock. Robust multi-body motion tracking using commute time clustering. In *Proc ECCV*, pages 160–173, 2006.
24. H. Qiu and E.R. Hancock. Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1873–1890, 2007.
25. A. Robles-Kelly and E. R. Hancock. String edit distance, random walks and graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):315–327, 2004.
26. A. Robles-Kelly and E. R. Hancock. Graph edit distance from spectral seriation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):365–378, 2005.
27. T. Rudolph. Constructing physically intuitive graph invariants. *http://arXiv.org/abs/quant-ph/0206068*, 2002.
28. L. Sarti, M. Maggini, and M. Gori. Exact and approximate graph matching using random walks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1100–1111, 2005.
29. N. Shenvi, J. Kempe, and K. B. Whaley. A quantum random walk search algorithm. *Phys. Rev. A*, 67(5), 2003.
30. S. Shiau, R. Joynt, and S. N. Coppersmith. Physically-motivated dynamical algorithms for the graph isomorphism problem. *Quantum Information and Computation*, 5:402–506, 2005.
31. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.
32. A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach.* Birkhauser Verlag, Boston, 1993.
33. Mario Szegedy. Quantum speed-up of markov chain based algorithms. *FOCS*, pages 32–41, 2004.
34. A. Webb. *Statistical Pattern Recognition, 2nd Ed.* John Wiley & Sons, Chichester, England, 2002.
35. T. Yamasaki, H. Kobayashi, and H Imai. Analysis of the absorbing times of quantum walks. *Phys. Rev. A*, 68:012302, 2003.
36. X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *ICML*, pages 1561–1566, 2003.