

Graph Edit Distance without Correspondence from Continuous-Time Quantum Walks

David Emms, Richard C. Wilson, and Edwin R. Hancock

Department of Computer Science, University of York, York, YO10 5DD, UK

Abstract. We consider how continuous-time quantum walks can be used for graph matching. We focus in detail on both exact and inexact graph matching, and consider in depth the problem of measuring graph similarity. We commence by constructing an auxiliary graph, in which the two graphs to be matched are co-joined by a layer of indicator nodes (one for each potential correspondence between a pair of nodes). We simulate a continuous time quantum walk in parallel on the two graphs. The layer of connecting indicator nodes in the auxiliary graph allow quantum interference to take place between the two walks. The interference amplitudes on the indicator nodes are determined by differences in the two walks. We show how these interference amplitudes can be used to compute graph edit distances without explicitly determining node correspondences.

1 Introduction

The problem of determining whether a pair of graphs are isomorphic has been extensively studied in the literature. Polynomial time algorithms have been given for graphs with a number of special properties [14,1,11]. Currently the best algorithm for isomorphism testing of general graphs is ‘Nauty’ developed by McKay [5]. Nauty constructs a canonical labelling of the vertices in a graph and finds generators of the graph’s automorphism group. For most graphs it is extremely fast but its complexity is exponential for some classes of graphs [15]. Another important algorithm that carries out subgraph isomorphism and clique detection, as well as isomorphism testing, was developed by Ullmann [17]. The algorithm uses back-tracking to successively eliminate nodes from the search space. When we wish to consider practical problems, however, we typically need to be able to deal with structural errors. For this reason, inexact graph matching must be carried out.

Quantum computing has recently attracted attention because of the potential for considerable speed-ups over classical algorithms. For instance, Grover’s search algorithm [10] is quadratically faster and Shor’s factorisation algorithm [16] is exponentially faster than known classical algorithms. However, quantum algorithms also have a richer structure than their classical counterparts since they use qubits rather than bits as the basic representational unit. This structure is exploited in Shor’s algorithm where the Fourier transform is used to locate factors of a composite number. However, although quantum computers potentially offer computational powerful tools for solving problems of a combinatorial nature, developing algorithms that take advantage of this power has proved elusive. This is particularly the case for problems involving graphs, and in particular the subgraph isomorphism problem.

One route is potentially furnished by Jozsa's algorithm which performs period finding and can be generalised to solving the hidden subgroup problem [12]. The hidden subgroup problem is of interest since there is a famous reduction of the graph isomorphism problem to it, see for example [12]. Beals was able to show how to efficiently carry out the quantum Fourier transform on the symmetric group [3] but, despite much attention, a quantum algorithm that uses this to efficiently solve the graph isomorphism problem has not been found. In fact, it seems unlikely that the graph isomorphism problem will succumb to a direct attack using the quantum Fourier transform on the symmetric group. In a recent paper, Kempe and Shalev [9] examine which subgroups of the symmetric group can be distinguished from the identity subgroup using the methods of quantum Fourier sampling. Ettinger and Høyer [8] propose a quantum observable for the graph isomorphism problem. However, the computational overheads are prohibitive.

In this paper we take a different approach, based on quantum walks. These have been introduced as quantum counterparts of random walks and a good summary is given by Kempe [13]. The behaviour of quantum walks is governed by unitary rather than stochastic matrices. Quantum walks provide an approach to designing quantum algorithms that lends itself more to physical intuition. By making use of the exponentially faster hitting times that are observed for continuous-time quantum walks on graphs [13], 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 [4].

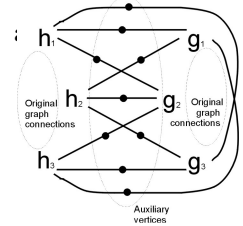
Our contribution in this paper is to develop a means of computing graph edit distance based on the continuous-time quantum walk. As we will show, it is also possible to construct the continuous-time quantum walk so that the destructive interference takes place continuously rather than requiring a final interference step. As the interference takes place continuously, the interference amplitudes that we observe are reliant more directly on the differences between the two walks at all times during the walk's evolution. Finally, the quantum amplitudes are complex-valued rather than real-valued thus giving a two dimensional distribution of interference amplitudes, and therefore more information with which to calculate the assignment probabilities. We show how to compute a measure of graph similarity using the interference amplitudes and without the need to determine explicit node correspondences. This represents a development of our recent work where we have a) used continuous time quantum walks to develop non-classical extension of the commute time [7] and b) used quantum interference to compute explicit correspondences [6].

2 The Continuous-Time Quantum Walk

The continuous-time quantum walk is a natural quantum analogue of the classical random walk. Like the classical random walk on a graph, its state space is the set of vertices of the graph. However, as with the discrete-time quantum walk, the probability of being at a certain state is given by the square of the amplitude of that state, rather than just the amplitude of the state (as is the case classically). As with the discrete-time quantum walk, 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 a vertex u , it moves to adjacent vertices 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, and unlike the discrete-time quantum walk where basis states correspond to arcs. 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 $|\psi_t\rangle = \sum_{u \in \mathcal{V}} \alpha_u(t) |u\rangle$. Thus, the amplitudes $\alpha_u(t) \in \mathbb{C}$.

The probability of the walk being in a particular state is given by the square of the amplitude of that state. Let X_t be the random variable giving the location of the walk at time t . The probability of being at $u \in V$ at time t is given by $\Pr(X^t = u) = \alpha_u \alpha_u^*$, 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 $u \in V, t \in \mathbb{R}^+$, and $\sum_{u \in V} \alpha_u(t) \alpha_u^*(t) = 1$ for all $t \in \mathbb{R}^+$. The evolution of the state vector is given by $\frac{d}{dt} |\psi_t\rangle = -iL |\psi_t\rangle$. Since the evolution of the probability vector of the walk at time t depends on the state vector of the walk (not merely the probability vector), we note that unlike the classical walk, the quantum walk is not a Markov chain. Given an initial state for the walk, $|\psi_0\rangle$, Equation 2 can be solved to give $|\psi_t\rangle = e^{-iLt} |\psi_0\rangle$. Thus, given the initial state, we can calculate the state of the walk at an time, t .



3 The Auxiliary Graph

Given two graphs, G and H , an auxiliary graph, $\Gamma(G, H)$, is constructed on which the walks can interfere. The auxiliary graph consists of the original two graphs, augmented by a set of auxiliary vertices that are used as correspondence indicators. Each auxiliary vertex is connected by an edge to one node in each of the original two graphs. That is, each pair of vertices, $g \in \mathcal{V}_G$ and $h \in \mathcal{V}_H$, one from each graph, are both adjacent to an auxiliary vertex labelled $v_{\{g,h\}}$. The auxiliary graph is symmetric with respect to interchanging its two arguments and with respect to permutations of the vertices in either graph. A schematic representation of its structure is shown in Figure 1.

Fig. 1. The auxiliary graph, $\Gamma(G, H)$, showing the vertices $g_1, g_2, g_3 \in \mathcal{V}_G$ and $h_1, h_2, h_3 \in \mathcal{V}_H$ connected by way of auxiliary vertices

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$, then $\Gamma(G, H) = (\mathcal{V}_\Gamma, \mathcal{E}_\Gamma)$ and the vertex and edge sets can be decomposed such that $\mathcal{V}_\Gamma = \mathcal{V}_G \cup \mathcal{V}_H \cup \mathcal{V}_A$ and $\mathcal{E}_\Gamma = \mathcal{E}_G \cup \mathcal{E}_H \cup \mathcal{E}_A$ where

$$\begin{aligned} \mathcal{V}_A &= \{v_{\{g_i, h_j\}} | g_i \in \mathcal{V}_G, h_j \in \mathcal{V}_H\} \text{ and,} \\ \mathcal{E}_A &= \{\{g_i, v_{\{g_i, h_j\}}\}, \{h_j, v_{\{g_i, h_j\}}\} | g_i \in \mathcal{V}_G, h_j \in \mathcal{V}_H\}. \end{aligned}$$

The auxiliary vertices, \mathcal{V}_A , serve to connect the two graphs and act as sites on which the interference takes place. The auxiliary graph is similar to the association graph [2].

However, information about the structure of the two graphs comes from incorporating the original graphs themselves rather than through the connections between the auxiliary vertices, as is the case for the association graph. We delay our explanation of how the auxiliary graph allows graph matching to be carried out until after we have provided the mathematical details of the discrete-time quantum walk on the auxiliary graph, which we do in the next section.

4 The Continuous-Time Quantum Walk on the Auxiliary Graph

The auxiliary graph, $\Gamma(G, H)$, based on the two graphs $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $H = (\mathcal{V}_H, \mathcal{E}_H)$ was described in the previous section. In this section we describe the simulation of the continuous-time quantum walk on the auxiliary graph. The state of the quantum walk on Γ at time t is given by $|\psi_t\rangle = \sum_{u \in \mathcal{V}_\Gamma} \alpha_u(t) |u\rangle$. The starting state has amplitudes

$$\alpha_u(0) = \begin{cases} \frac{d(u)}{C} & \text{if } u \in \mathcal{V}_G; \\ -\frac{d(u)}{C} & \text{if } u \in \mathcal{V}_H; \\ 0 & \text{otherwise;} \end{cases}$$

where $C = \sum_{u \in \mathcal{V}_H \cup \mathcal{V}_G} d(u)^2$ is the normalisation constant such that the probabilities sum to unity. This starting state is used rather than the more natural starting state

$$\tilde{\alpha}_u(0) = \begin{cases} \frac{1}{C} & \text{if } u \in \mathcal{V}_G; \\ -\frac{1}{C} & \text{if } u \in \mathcal{V}_H; \\ 0 & \text{otherwise;} \end{cases}$$

where $\tilde{C} = \sqrt{|\mathcal{V}_G||\mathcal{V}_H|}$, since $|\tilde{\psi}_0\rangle = \sum_{u \in \mathcal{V}_\Gamma} \tilde{\alpha}_u(0) |u\rangle$ is an eigenvector of the Laplacian for Γ and thus a stationary state of the walk. If we were to use this starting state then all the interference amplitudes would remain zero at all times and no information concerning the two walks would be obtained.

4.1 Behaviour of the Walk

We now consider the behaviour of the walk on the auxiliary graph, taking separately the cases when the graphs being compared are isomorphic and non-isomorphic.

Isomorphic Graphs. Let G and H be two graphs such that $\zeta : \mathcal{V}_G \rightarrow \mathcal{V}_H$ is an isomorphism between them.

Theorem 1. *If $\zeta : \mathcal{V}_G \rightarrow \mathcal{V}_H$ is a graph isomorphism such that $h = \zeta(g)$ then $\alpha_g(t) = -\alpha_h(t)$.*

Proof. Let L be the Laplacian matrix for $\Gamma = \Gamma(G, H)$ and \tilde{L} be the Laplacian for $\tilde{\Gamma} = \Gamma(H, G)$. The basis states for the walk on Γ are $|u\rangle$ such that $u \in \mathcal{V}_\Gamma$ and those on $\tilde{\Gamma}$ are $|\tilde{u}\rangle$ such that $u \in \mathcal{V}_{\tilde{\Gamma}}$.

Let \mathcal{P} be the permutation

$$\mathcal{P}|u\rangle = \begin{cases} |\tilde{\zeta}(u)\rangle & \text{if } u \in \mathcal{V}_G; \\ |\tilde{\zeta}^{-1}(u)\rangle & \text{if } u \in \mathcal{V}_H; \\ |\tilde{v}_{\{\zeta^{-1}(h), \zeta(g)\}}\rangle & \text{if } u \in \mathcal{V}_G \\ & \text{and } u \sim g \in \mathcal{V}_G, u \sim h \in \mathcal{V}_H. \end{cases}$$

The permutation \mathcal{P} is such that $\tilde{L} = \mathcal{P}L\mathcal{P}^T$. The starting state for the walk on Γ is $|\psi_0\rangle = \frac{1}{C}(\sum_{u \in \mathcal{V}_G} |u\rangle - \sum_{v \in \mathcal{V}_H} |v\rangle)$ and the starting state for the walk on $\tilde{\Gamma}$ is $|\tilde{\psi}_0\rangle = \frac{1}{C}(\sum_{u \in \mathcal{V}_H} |u\rangle - \sum_{v \in \mathcal{V}_G} |v\rangle)$. We note that

$$\mathcal{P}^{-1}|\tilde{\psi}_0\rangle = \frac{1}{C}\mathcal{P}^{-1}(\sum_{u \in \mathcal{V}_H} |u\rangle - \sum_{v \in \mathcal{V}_G} |v\rangle) = \frac{1}{C}(\sum_{u \in \mathcal{V}_G} |u\rangle - \sum_{v \in \mathcal{V}_H} |v\rangle) = -|\psi_0\rangle \quad (1)$$

Consider the amplitude $\alpha_g(t)$:

$$\alpha_g(t) = \langle \tilde{g} | e^{-i\tilde{L}t} | \tilde{\psi}_0 \rangle = \langle \tilde{g} | e^{-i\mathcal{P}L\mathcal{P}^{-1}t} | \tilde{\psi}_0 \rangle = \langle \tilde{g} | \mathcal{P}e^{-iLt}\mathcal{P}^{-1} | \tilde{\psi}_0 \rangle = -\langle h | e^{-iLt} | \psi_0 \rangle = -\alpha_h(t) \quad (2)$$

□

Theorem 2. *If $\zeta : \mathcal{V}_g \rightarrow \mathcal{V}_h$ is a graph isomorphism such that $h = \zeta(g)$ then $\alpha_{v_{\{g,h\}}}(t) = 0$ for all $t \in \mathbb{R}^+$.*

Proof. Let $x(t) = \alpha_{v_{\{g,h\}}}(t)$. Equation 2 gives $d/dt|\psi_t\rangle = -i\mu L|\psi_t\rangle$. Thus, $d/dtx(t) = -i(\alpha_h(t) + \alpha_g(t) - 2x(t)) = 2i\mu x(t)$, by Theorem 1. The general solution is $x(t) = A_1 e^{2i\mu t}$. Since $x(0) = 0$ we have that $A_1 = 0$, and thus $\alpha_{v_{\{g,h\}}}(t) = x(t) = 0$ for all $t \in \mathbb{R}^+$. □

Non-isomorphic Graphs. We now consider the case when G and H are non-isomorphic, but closely related graphs. In this case, in general, $\alpha_{v_{\{g,h\}}}(t) \neq 0$ even if $\langle g, h \rangle$ is a true assignment. However, the algorithm relies upon $\alpha_{v_{\{g,h\}}}(t)$ being smaller in magnitude for true assignments than it is for false assignments. We investigate whether this assumption allows successful matching to be carried out in Section 6. In the next section we consider at what point in time the interference amplitudes for the walk should be used so that these amplitudes are most easily distinguished.

4.2 Measuring the Interference Amplitudes

The algorithm that we propose uses a set of interference amplitudes, corresponding to the set of possible vertex-vertex assignments, in order to carry out graph matching. For exact graph matching the point during the evolution of the walk at which we use the interference amplitudes, $\alpha_{v_{\{g,h\}}}(\tau)$, $g \in \mathcal{V}_G$, $h \in \mathcal{V}_H$, is not important. We know that $\alpha_{v_{\{g,h\}}}(t) = 0$ for all $t \in \mathbb{R}^+$ if $\langle g, h \rangle$ is a true assignment, and it is highly improbable that if $\langle g, h \rangle$ is a false assignment then $\alpha_{v_{\{g,h\}}}(\tau)$ will be exactly at some particular τ but not at other times. Thus, we base our choice of τ on how well it will allow inexact graph matching to be carried out.

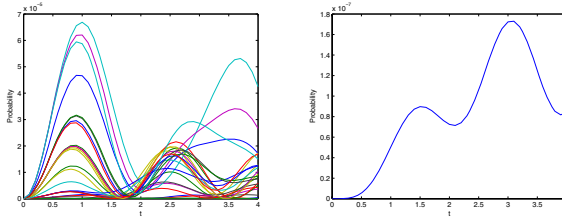


Fig. 2. The probabilities associated with the interference states for a particular vertex (left) and for just the interference state for the true assignments (right). Note the difference in scale on the y-axis.

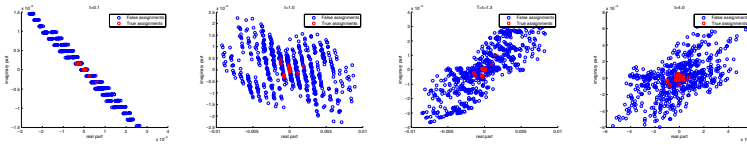


Fig. 3. The distributions of the interference amplitudes for a pair of non-isomorphic graphs, G and H , at $t = 0.1$, $t = 1.0$, $\tau = t = 1.3$ and $t = 4.0$. The probabilities for the interference states associated with a particular vertex $g \in \mathcal{V}$ are shown in Figure 2.

Let G and H be a pair of non-isomorphic, related graphs. The probabilities for the set of $|\mathcal{V}_H|$ interference states associated with a particular vertex $g \in \mathcal{V}_G$ are shown in Figure 2. The distribution of the complete set of interference amplitudes at various times during the evolution of the walk are shown in Figure 3. We note that the probabilities for the false assignments have similar periods. In comparison, the probability for the true assignment continues to increase with time as the differences in the states of the two walks increase. Thus we select τ such that the variance of the real part of the distribution of all interference amplitudes is at its first maximum. This corresponds to $t = 1.3$ in Figures 2 and 3.

We chose this value of τ since at this time the interference amplitudes for the false matches are all close to their local maxima. Thus the separation between the true and false interference amplitudes should be at its greatest. Since the periods for the probabilities for the false assignments are not exactly the same, the subsequent maxima are not as close in terms of the times at which they are obtained. Additionally, as t increases the probabilities for the true assignments continue to increase, thus they are most easily distinguished at the first maximum. We will henceforth refer to $\alpha_{v_{\{g,h\}}}(T)$ as *the* interference amplitude for $g \in \mathcal{V}_G$ and $h \in \mathcal{V}_H$ and will denote it by $\alpha^-(g, h)$, or simply α if it is unimportant to which pair of vertices it refers.

5 A Distance Measure from the Interference Amplitudes

Let $\Omega = \{|\alpha^-(g_i, h_j)|^2 : g_i \in \mathcal{V}_G, h_j \in \mathcal{V}_H\}$ be the squared magnitudes of the set of interference amplitudes for the two graphs. Further, let $\Omega_{|\mathcal{V}_G|}$ (where G is the graph

with the fewest amplitudes if $|\mathcal{V}_G| \neq |\mathcal{V}_H|$) be the smallest $|\mathcal{V}_G|$ elements of Ω . We define $d(G, H)$ to be the variance of $\Omega_{|\mathcal{V}_G|}$ divided by the variance of Ω :

$$d(G, H) = \frac{\text{var}(\Omega_{|\mathcal{V}_G|})}{\text{var}(\Omega)}.$$

Theorem 3. *If $G \cong H$ then $d(G, H) = 0$*

Proof. If $G \cong H$ then there exists an isomorphism, $\zeta : \mathcal{V}_G \rightarrow \mathcal{V}_H$. Thus, for each $g \in \mathcal{V}_G$ we have, by Theorem 2, $\alpha^-(g, \zeta(g)) = 0$, for all $t \in \mathbb{R}^+$. Therefore there are at least $|\mathcal{V}_G|$ assignment amplitudes equal to zero, hence $\text{var}(\Omega_{|\mathcal{V}_G|}) = 0$ and so $d(G, H) = 0$. \square

Note that we only need calculate the set of interference amplitudes in order to calculate $d(G, H)$, we need not compute the correspondence match between the graphs. As we will show in the experimental section, the variance of the smallest interference amplitudes are closely related to the edit distance between graphs.

6 Experiments

We now consider the proposed graph distance measure, firstly on synthetic graphs and secondly on graphs obtained from real-world data. Figure 4 shows plots of the distance measure as functions of the amount of noise. The left-hand panel in Figure 4 shows the distance measure averaged over the graphs with different numbers of vertices. It can be seen that the relationship between the distance measure and the amount of noise is close to linear. The right-hand panel in Figure 4 shows the distance measure as a function of noise for graphs with different numbers of vertices. The distance d increases steadily as the level of noise is increased for all the graphs.

We now explore the use of the distance measure for graphs derived from real world image data. For this we take images from the COIL database and construct Delaunay triangulations. We consider 12 different objects from the database with view angles between 0° and 45° . Firstly, we consider the distance measure's ability to form trajectories of the graphs, ordering them according to the view angle. Secondly, we analyse the effectiveness of the distance measure for graph clustering. Examples of views of each of the images used together with their Delaunay triangulations are shown in Figure

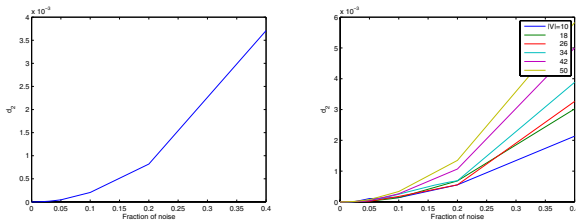


Fig. 4. [Left: The distance measure as a function of the amount of noise averaged over graphs with between 10 and 50 vertices. Right: As a function of the amount of noise for graphs with between 10 and 50 vertices.

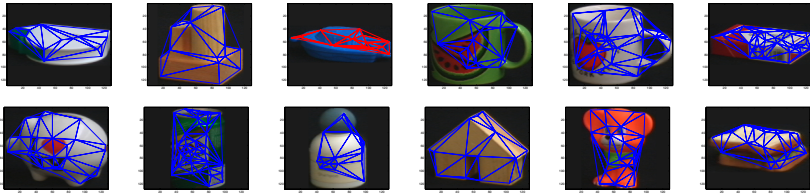


Fig. 5. Example COIL Images

5. Note that the images are represented by un-attributed, un-weighted graphs—all the information is represented in the connectivity structure of the Delaunay triangulations.

Figure 6 shows the average of our graph distance as a function of the difference in angle between the views from which the graphs were obtained. The distance between a graph and itself (that is, when the angular distance is zero) is guaranteed to be zero by Theorem 3. The difference in the distance measure between a pair of graphs with zero angular distance (between the images) and 5° is significantly larger than any other change in distance for a 5° change in angular distance. This shows that the distance measure distinguishes most strongly between pairs of isomorphic graphs and pairs of graphs that are related but non-isomorphic than it does between closely related and distantly related non-isomorphic graphs. Such behaviour reflects the understanding that there is a clear distinction between isomorphic graphs and non-isomorphic graphs but a less clear distinction (and hence a smaller change in graph distance) between ‘closely’ and ‘less closely’ related graphs.

We now turn our attention to the ability of the distance measure to distinguish between different sets of graphs and so allow graph clustering to be carried out. We look at the ability of the distance measure to cluster two distinct sets of graphs. That is, we take two sets of graphs at a time, carry out dimensionality reduction on the matrix of distances between them and embed them in 2D space. We then look at whether the two clusters are linearly separable, and what proportion of the graphs in each set have another graph from the same set as their nearest neighbour.

We take two sets of graphs, each corresponding to views of a different object, each set containing 10 graphs. We form the matrix of distances between all these graphs to give a 20 by 20 matrix of distances. We then perform MDS on this matrix of distances to produce a 2D embedding of the graphs, where one data point corresponds to one graph. Table 1 shows the number of times the clusters of graphs for each of the 132 pairings of the objects are linearly separable when the two sets of graphs are embedded using MDS. It also shows the proportion of times that the nearest neighbour for a particular graph in the cluster is a graph derived from the same image as it was. Looking at the second column, we see that in over half of the cases the two clusters are linearly separable. The third column shows the total number of graphs in each set for which its nearest

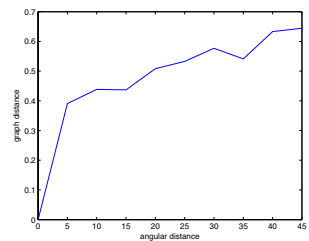


Fig. 6. Distance Measure versus Angular Distance for COIL Graphs

Table 1. Clustering results of graphs derived from images from COIL. The number of clusters that are linearly separable and the mean number of graphs whose nearest neighbour is from the same set, averaged over all of the clusters.

| Object | Number of clusters linearly separable (out of 11) | Nearest neighbour from same set (out of 10) |
|--------|--|--|
| 1 | 6 | 9.1 |
| 2 | 7 | 8.5 |
| 3 | 4 | 8.4 |
| 4 | 4 | 7.9 |
| 5 | 7 | 9.2 |
| 6 | 8 | 9.5 |
| 7 | 7 | 8.9 |
| 8 | 10 | 9.7 |
| 9 | 3 | 8.2 |
| 10 | 6 | 8.7 |
| 11 | 6 | 8.8 |
| 12 | 6 | 8.7 |
| Mean | 6.2 | 8.8 |

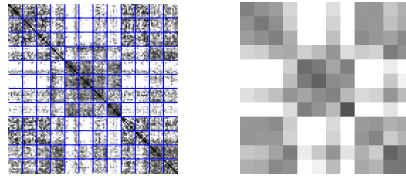


Fig. 7. The matrix of distances between individual graphs (left) and the average distances between sets of graphs (right). Completely black indicates a distance of 0, the lighter the grey the greater the distance.

neighbour is from the same set, averaged over the 11 clusterings carried out for each object. The mean for all the different sets is 8.8 out of a possible 10 indicating that the distance measure distinguishes successfully between graphs from different sets.

Figure 7 shows the matrix of distances between pairs of graphs and the mean distances between the sets of graphs. The smallest distances between graphs (left-hand plot, those closest to black in the figure) lie closest to the diagonal, these are the pairs of graphs representing images of the same objects and with a small angular distance between them.

7 Conclusions

We have suggested a distance measure based on the interference of the continuous-time quantum walk. The measure is computed by taking the variance of the smallest n interference amplitudes (where n is the number of vertices) and normalising by dividing by the variance of the whole distribution. Our experiments showed that this measure performed well.

References

1. Babai, L., Grigoryev, D.Y., Mount, D.M.: Isomorphism of graphs with bounded eigenvalue multiplicity. In: STOC 1982: Proceedings of the fourteenth annual ACM symposium on Theory of computing, pp. 310–324. ACM Press, New York (1982)
2. Barrow, H.G., Burstall, R.M.: Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters* 4(4), 83–84 (1976)
3. Beals, R.: Quantum computation of Fourier transforms over symmetric groups. In: Proceedings of the 29th Annual ACM Symposium on the Theory of Computation (STOC), Texas, pp. 48–53. ACM Press, New York (1997)
4. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: STOC 2003: Proc. 35th ACM symposium on TOC, pp. 59–68. ACM Press, New York (2003)
5. McKay, B.D.: Practical graph isomorphism. *Congressus Numerantium* 30, 45–87 (1981)
6. Emms, D., Hancock, E.R., Wilson, R.C.: Graph similarity using interfering quantum walks. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 823–831. Springer, Heidelberg (2007)
7. Emms, D., Wilson, R.C., Hancock, E.R.: Graph Embedding Using Quantum Commute Times. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 371–382. Springer, Heidelberg (2007)
8. Ettinger, M., Høyer, P.: A quantum observable for the graph isomorphism problem (1999)
9. Ettinger, M., Høyer, P.: The hidden subgroup problem and permutation group theory (2004)
10. Grover, L.: A fast quantum mechanical algorithm for database search. In: Proc. 28th Annual ACM Symposium on the Theory of Computation, pp. 212–219. ACM Press, New York (1996)
11. Hopcroft, J.E., Wong, J.K.: A linear time algorithm for isomorphism of planar graphs. In: Proceedings of the 6th Annual ACM Symposium on the Theory of Computing (STOC 1974), pp. 172–184 (1974)
12. Jozsa, R.: Quantum factoring, discrete logarithms, and the hidden subgroup problem. *Computing in Science and Engineering* 03(2), 34–43 (2001)
13. Kempe, J.: Quantum random walks—an introductory overview. *Contemporary Physics* 44(4), 307–327 (2003)
14. Köbler, J.: On graph isomorphism for restricted graph classes. In: Beckmann, A., Berger, U., Löwe, B., Tucker, J.V. (eds.) CiE 2006. LNCS, vol. 3988, pp. 241–256. Springer, Heidelberg (2006)
15. Miyazaki, T.: The complexity of McKay’s canonical labeling algorithm (1996)
16. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26, 1484–1509 (1997)
17. Ullmann, J.R.: An algorithm for subgraph isomorphism. *J. ACM* 23(1), 31–42 (1976)