This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Graph Isomorphisms and Automorphisms via Spectral Signatures

Dan Raviv, Ron Kimmel *Fellow, IEEE,* and Alfred M. Bruckstein

Abstract—An isomorphism between two graphs is a connectivity preserving bijective mapping between their sets of vertices. Finding isomorphisms between graphs, or between a graph and itself (automorphisms) is of great importance in applied sciences. The inherent computational complexity of this problem is yet unknown. Here, we introduce an efficient method to compute such mappings using heat kernels associated to the graph Laplacian. While the problem is combinatorial in nature, in practice we experience polynomial runtime in the number of vertices. As we demonstrate, the proposed method can handle a variety of graphs, and is competitive with state of the art packages on various important examples.

Index Terms—Graph isomorphism, Graph symmetries, Graph automorphisms, Graph Laplacian, Heat kernel maps, Heat kernel signatures

1 Introduction

A one to one mapping between the vertex sets of two given graphs, such that connectivity is preserved, is called an isomorphism or graph-isometry. Mapping a graph to itself in a similar structure preserving manner is an automorphism or graph-symmetry. There is no known polynomial-time algorithm for finding such mappings, and the problem was never classified as NP-complete. The graphs can be directed or undirected, weighted or unweighted, and possibly even disconnected. Here, we limit our discussion to the problem of graph symmetry/isometry extraction for undirected, weighted and unweighted, connected graphs.

Symmetries and isometries of graphs play an important role in modern science. In chemistry for example, symmetries can predict chemical properties of a given material [1], as molecules can be classified according to symmetries of the graph representing the connectivity between their atoms.

Babai and Lukas' paper [2] on permutation groups, provided an upper bound of $\exp(\sqrt{n \log n})$ for finding graph symmetry/isometry, where n is the number of vertices in the graph. Restricting the structure of the graph better bounds were found. Such restrictions involve limiting the degree of vertices [3], or consideration of hyper graphs of fixed rank [4]. For some special type of graphs even linear complexity was proven. Such graphs include interval graphs [5], planar graphs [6], and graphs with bounded eigenvalue multiplicity [7].

Treating either very simple types of graphs or dealing with exponential complexity poses a challenge for applied sciences. Heuristic approaches for general graphs have been proposed and were found to be quite efficient in many practical applications. Some, e.g. [8], [9] suggested to use the branch-and-bound approach, which is an exhaustive search algorithm with pruning that can be applied to graphs with a small number of vertices. Gori *et al.* [10] experimented with random walks, while Umeyana [11] investigated the eigen-decomposition of the adjacency matrix. Several fast canonical labeling algorithms were proposed to address the graph-isometry problem, such as Ullmann's algorithm [12], VF [13] and VF2 [14]. In addition, software packages implementing fast labeling such as *Bliss* [15], *Nauty* [16] and *Saucy* [17] are well known. These tools can detect isometries and symmetries for graphs with tens of thousands of vertices quite efficiently for many different graphs.

Isometries of shapes can sometimes be translated to isomorphisms of graphs. Bèrard *et al.* [18] considered embedding of Riemannian manifolds into an infinite dimensional Euclidean space defined by the eigenfunctions of the Laplace Beltrami operator in order to compute the Gromov-Hausdorff distance between such geometric structures. Rustamov [19] applied this idea to surface matching. Horaud *et al.* [20] proposed a matching process based on the eigenvectors of the Laplace Beltrami operator, Sun *et al.* [21] noted that the diagonal of the heat kernel is a stable shape descriptor when evaluated in several scales, while Ovsjanikov *et al.* [22] used heat kernels to find correspondences between shapes, and Xiao *et al.* [23] discussed the structure of graphs as reflected in the heat kernel trace. This research led to a variety of algorithms which define and search approximate symmetries and isometries [24], [9] between two and high dimensional shapes.

This paper was motivated by the Ovsjanikov *et al.* paper [22] on isometries between surfaces. They discussed structures with one possible symmetry which lead towards a simple (one point) matching algorithm based on heat distribution. In this note we consider shapes (graphs) with many automorphisms and the ambiguity

• *Department of Computer Science, Technion, Israel Institute of Technology. E-mail: {darav,ron,freddy}@cs.technion.ac.il*

Digital Object Indentifier 10.1109/TPAMI.2012.260 0162-8828/12/$31.00 © 2012 IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

of their heat kernel maps.

We provide theoretical support for the uniqueness of the signatures and justify the fact that a subset of matching vertices is sufficient for solving the problem as a whole. We then propose a greedy algorithm for handling signatures in a process of finding isometries and symmetries, which is exponential in the worst case, yet appears to be linear (in the number of symmetries) in practice.

The rest of the paper is organized as follows; Section 2 reviews three definitions of graph Laplacians, followed by Section 3 where the heat kernel signatures are defined and discussed. Section 4 is devoted to a method for evaluating spectral signatures, and Section 6 describes the proposed isomorphism computation algorithm. We provide numerical validation in Section 7 and conclude in Section 8.

2 Graph Laplacian

A graph $G = (V, E)$ is defined as a set of vertices V and edges $E \subseteq V \times V$ describing the vertex connectivity. In this note we consider G to be undirected, connected, without trivial loops. We define the symmetric adjacency matrix A by

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

and the diagonal matrix $D(u,u) = \deg(u)$ $u \in V$ displaying the vertices' degrees.

In the literature there are two alternative definitions for graph Laplacians, a *standard* and a *normalized* Laplacian [25]. The standard Laplacian is defined as

$$L = D - A, \quad (2)$$

while the normalized Laplacian is given by

$$\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \quad (3)$$

Both Laplacians are positive semidefinite, and, hence, have nonnegative eigenvalues. The normalized version's eigenvalues are bounded by 2 from above, but both are adequate for our framework.

A weighted Laplacian can also be defined, which we shall use in evaluating approximate symmetries, also known as ϵ-symmetries [24]. In this case the adjacency matrix is defined as

$$\tilde{A}(u,v) = \begin{cases} w(u,v) & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

and the diagonal matrix becomes $\tilde{D}(u,u) = \sum\limits_{(u,v)\in E} w(u,v)$. The weighted Laplacian is defined as before

$$\tilde{L} = \tilde{D} - \tilde{A}. \quad (5)$$

The weights $w(u,v)$ for graphs with vertices embedded in a metric space can be computed using l_2 or l_1 distances between the spatial location of the vertices.

3 Heat kernel signatures

One way to analyze graphs is based on heat flows. In nature heat diffusion is governed by the *heat equation*

$$\left(\Delta + \frac{\partial}{\partial t}\right) f(x;t) = 0, \quad (6)$$

where Δ represents the continuous Laplace-Beltrami operator, and $f : X \times \mathbb{R}^+ \to \mathbb{R}$ a time varying scalar function on the manifold X. Substantial research in geometry was done to analyze the heat equation in general, and specifically the Laplace Beltrami operator. One branch of modern shape analysis focuses on spectral properties of the Laplacian operator to address problems from shape matching to shape retrieval. Here, we follow this line of research in the discrete domain of graphs.

The heat kernel, which is the impulse response solution of (6), describes the heat flow between vertices, and can be evaluated from the spectral decomposition of the Laplacian [25]

$$K_t(x,y) = \sum_{l=0}^{|V|} e^{-\lambda_l t} \phi_l(x) \phi_l(y), \quad (7)$$

where λ_l, ϕ_l are the eigenvalues and eigenfunctions of the Laplacian, and $x, y \in V$. As we consider symmetric Laplacian such decomposition always exists.

In shape analysis, a special attention was given to the diagonal of the heat kernel $K_t(x,x)$. Sun *et al.* [21] introduced a robust local shape descriptor referred to as *heat kernel signature* (HKS), that is evaluated from the heat propagation at different scales. In addition to the diagonal, additional information can be extracted from the rows of the kernel. A vertex q at time t, defines a map from the vertices of a graph to $\mathbb{R}$ by considering the mapping $K_t(q,\cdot) : X \to \mathbb{R}$, known as a *heat kernel map* [22]. These maps play a major role in the forthcoming construction.

4 Spectral signatures

In what follows we build a unique descriptor for each vertex in the graph (V, E) based on the eigendecomposition of the Laplacian, and a subset of k graph vertices.

We define a k-signature $\mathcal{S}^k(u)$ for a vertex u based on k chosen vertices $\{p^i\}_{i=1}^k$ and $|T|$ times $\{t_1, t_2, ..., t_{|T|}\}$, to be the vector of length $|T| \times k$

$$\mathcal{S}^k(u) = \left(K_t(p^i, u)\right)_{i=1}^k \quad t \in T, \quad (8)$$

where we concatenate all kernel values to one column signature.

We shall show that for every undirected, connected graph, there exists a subset of vertices $\{p^i\}_i$ which defines a unique signature $\mathcal{S}^k(u)$ for every vertex given $|V|$ times are used, meaning that:

$$\mathcal{S}^k(u) = \mathcal{S}^k(v) \to u = v, \quad (9)$$

and, as shown in the next section, this signature is also unique for isomorphic graphs. In some cases $k = |V|$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

chosen vertices are needed, for examples in cliques, but surprisingly in many instances much fewer vertices are required and this value depends on the number of repeated eigenvalues and the values of the corresponding eigenvectors themselves.

If all eigenvalues are distinct then inferring that the signatures are bijective can be done given one vertex assuming its value is not zero in all eigenvectors, as can be seen in Theorem 1. A more general result is given in Theorem 2 where more vertices are needed for constructing distinct signatures.

Lemma 1: Assuming $\lambda_i \in \mathbb{R}$ are distinct, $a, b \in \mathbb{R}^k$, then $\Sigma_{i=1}^k \exp(-\lambda_i t) a_i = \Sigma_{i=1}^k \exp(-\lambda_i t) b_i$ for every t if and only if $a_i = b_i$ for all i.
Proof: If $a_i = b_i$ we clearly have equality. Choosing k different times allows us to write k different equations, one for each t, using an invertible matrix M such that $Ma = Mb$, where $M \in \mathbb{R}^{k\times k}$, and $M_{ij} = \exp(-\lambda_j t_i)$. We conclude that $a_i = b_i$ for all i. □.

Lemma 2: Given $\{\phi_i\}_{i=1}^n$ eigenfunctions of the Laplacian. $\phi_i(u) = \phi_i(v)$ for all i if and only if $u = v$.
Proof: In one direction the proof is trivial. Consider a matrix where column i is the vector ϕ_i, then $\phi_i(u)$ for all i is a row vector in that matrix. The matrix is invertible as its columns are linear independent, hence, its rows must be linear independent as well. Because two rows can not be same, $\phi_i(u)$ for all i is a distinct row vector for every u. □

Theorem 1: If the Laplacian does not have repeated eigenvalues and vanishing values in its eigenvectors, then there exists one vertex p for which $\mathcal{S}^1(u)$ is distinct for every u. In other words, $\mathcal{S}$ is bijective.
Proof: From Lemmas 1 and 2, given $a_i = \phi_i(p)\phi_i(u)$, it follows that $\phi_i(p)\phi_i(u) = \phi_i(p)\phi_i(v)$ for all i. Since $\phi_i(p) \neq 0$ we conclude from Lemma 1 that $u = v$. □

Theorem 2: For a general graph G there exist $k < n$ vertices for which $\mathcal{S}^k(u)$ is bijective.
Proof: For k vertices $\{p^i\}_{i=1}^k$ and n different times $\{t_j\}_{j=1}^n$ we assume $K_t(p^i, u) = K_t(p^i, v)$, which can be written as

$$\begin{bmatrix} M_{11}\phi_1(p^k) & M_{12}\phi_2(p^k) & \cdots & M_{1n}\phi_n(p^k) \\ M_{21}\phi_1(p^k) & M_{22}\phi_2(p^k) & \cdots & M_{2n}\phi_n(p^k) \\ \vdots & & & \\ M_{n1}\phi_1(p^k) & M_{n2}\phi_2(p^k) & \cdots & M_{nn}\phi_n(p^k) \end{bmatrix} = M^k$$

$$\begin{bmatrix} M^1 \\ M^2 \\ \vdots \\ M^k \end{bmatrix} \times \begin{bmatrix} \phi_1(u) - \phi_1(v) \\ \phi_2(u) - \phi_2(v) \\ \vdots \\ \phi_n(u) - \phi_n(v) \end{bmatrix} = \bar{0}, \tag{10}$$

where $M_{ij} = \exp(-\lambda_j t_i)$. While every M^i is not necessary invertible, we can extract n independed rows from their concatenation in (10) because for each j exists p such that $\phi_j(p) \neq 0$ and the different times in M^k (for all k) are chosen to prevent linear dependencies between the rows. Note that at most $n-1$ vertices are required to construct n linear independent rows since the Laplacian has one constant eigenvector there must be a row with two non vanishing coefficents. Finally, we conclude that $u = v$ using Lemma 2. □

5 From Automorphisms to Isomorphisms

In contrast to automorphisms, where one Laplacian decomposition was required to construct signatures, we now face two sets of eigenvalues and eigenvectors. Assuming G and $\tilde{G}$ are isomorphic ensures that the eigenvalues are equal, but the eigenfunctions can be chosen arbitrary in each subspace corresponding to repeated eigenvalues. In what follows we assume that there exists a decomposition of the Laplacian of G into λ_i eigenvalues and ϕ_i eigenvectors, and the Laplacian of $\tilde{G}$ into $\tilde{\lambda}_i$ and $\tilde{\phi}_i$ such that $\lambda_i = \tilde{\lambda}_i$ and $\phi_i = \tilde{\phi}_i$ for all i, where the equality in the last equation reads that there exists $f : G \to \tilde{G}$ such that $\phi_i(u) = \tilde{\phi}_i(f(u))$ for all u and for all i. Since the signatures remain the same for every choice of basis we only need to compensate for the reordering function f.

Lemmas 1 and 2, are technical results that will be useful here as well, while the uniqueness of the signatures needs to be redefined.

Given two isomorphic graphs G and $\tilde{G}$ and the corresponding eigendecomposition of their Laplacians λ_i, ϕ_i and $\tilde{\lambda}_i, \tilde{\phi}_i$, we define their k-signatures for $u, v \in G$ and $f : G \to \tilde{G}$ as before

$$\begin{aligned} \mathcal{S}^k(u) &= \left(K_t(p^i, u)\right)_{i=1}^k \quad t \in T \\ \tilde{\mathcal{S}}^k(f(v)) &= \left(K_t(\tilde{p}^i, f(v))\right)_{i=1}^k \quad t \in T, \end{aligned} \tag{11}$$

where $\tilde{p}^i = f(p^i)$ and p^i are the anchor vertices.

We will show that the signatures are unique in the sense that

$$\mathcal{S}^k(u) = \tilde{\mathcal{S}}^k(f(v)) \to u = v. \tag{12}$$

Signatures within each graph are unique as seen earlier. What remains to be shown is that between two isomorphic graphs the signatures are still bijective.

Theorem 3: If the Laplacians of the isomorphic graphs G and $\tilde{G}$, do not have repeated eigenvalues and vanishing values in their eigenvectors, then there exists a vertex p for which $\mathcal{S}^1(u)$ is distinct for every $u \in G$, and it corresponds to only one vertex in $\tilde{G}$.
Proof: $\mathcal{S}^1(u)$ is unique for every $u \in G$ as proven in Theorem 1. Following lemmas 1 and 2 we conclude that

$$\phi_i(p)\phi_i(u) = \tilde{\phi}_i(f(p))\tilde{\phi}_i(f(v)) \ \forall\ i, \tag{13}$$

where $f : G \to \tilde{G}$. Because $\phi(p) = \tilde{\phi}_i(f(p))$, and $\phi(v) = \tilde{\phi}_i(f(v))$ for every v, it follows that

$$\phi_i(p)\phi_i(u) = \phi_i(p)\phi_i(v) \ \forall\ i, \tag{14}$$

Algorithm 1: Greedy Algorithm for automorphisms (isomorphisms) evaluation

Input: Eigenfunctions and Eigenvalues of the graph (graphs)
Output: Set of possible automorphisms (isomorphisms) $\{\Phi\}$

Choose p^1 arbitrary and find all possible $(\tilde{p}^1)$ matches according to similarity of the Heat Kernel Signatures (HKS)
for $i > 1$ **do**
  Construct $\mathcal{S}^{i-1}(u)$ given $(p^j)_{j=1}^{i-1}$ and $\tilde{\mathcal{S}}^{i-1}(v)$ given $(\tilde{p}^j)_{j=1}^{i-1}$ for all u and v.
  For unique match $\mathcal{S}^{i-1}(u) = \tilde{\mathcal{S}}^{i-1}(v)$ define $\Phi(u) = v$.
  If all vertices are matched then an automorphism (isomorphism) is found. Add it to $\{\Phi\}$.
  If u is already matched to a different location or does not have any possible match then stop the search in this branch.
  For one (arbitrary) u such that $\mathcal{S}(u) = \tilde{\mathcal{S}}(v^j)$ $1 > j \geq k$ (e.g. exist k options) split the search and define $p^{i+1} = u$, and $\tilde{p}^{i+1} = v^j$.

Algorithm 2: Optimistic algorithm for one isomorphism evaluation

Input: Eigenfunctions and eigenvalues of the graphs
Output: One of possible isomorphism Φ

Choose p^1 arbitrary and find all possible $(\tilde{p}^1)$ matches according to similarity of the Heat Kernel Signatures (HKS).
for $i > 1$ **do**
  Construct $\mathcal{S}^{i-1}(u)$ given $(p^j)_{j=1}^{i-1}$ and $\tilde{\mathcal{S}}^{i-1}(v)$ given $(\tilde{p}^j)_{j=1}^{i-1}$ for all u and v .
  For unique match $\mathcal{S}(u) = \tilde{\mathcal{S}}(v)$ define $\Phi(u) = v$.
  If all vertices are matched then an isomorphism is found.
  If u is already matched to a different location or does not have any possible match then stop the search (no isometry found).
  For one (arbitrary) u such that $\mathcal{S}(u) = \tilde{\mathcal{S}}(v)$ define $p^{i+1} = u$, and $\tilde{p}^{i+1} = v$.

from which we conclude, as done in Theorem 1, that $u = v$, meaning $\mathcal{S}^1(u)$ is unique for every $u \in G$, and fits only one vertex in $\tilde{G}$. □

Theorem 4: For two general graphs G and $\tilde{G}$ there exists $k < n$ vertices for which $\mathcal{S}^k(u)$ is distinct for every $u \in G$, and has a unique match $\tilde{\mathcal{S}}^k(f(v))$ in $\tilde{G}$.
Proof: The proof is identical to that of Theorem 2, using the correspondence function f such that

$$\phi_i(u) = \tilde{\phi}_i(f(v)) \;\; \forall \; i. \tag{15}$$

6 ALGORITHMS

From theorems 2 and 4 we conclude that only a subset of vertices is required to construct unique signatures and hence define an automosrphism or an isomorphism. We provide a greedy algorithm that constructs the signatures by adding new matches $p^i \rightarrow \tilde{p}^i$ in the i'th step. We must emphasize that even though we considered a joined eigendecomposition in the proof given earlier, it does not have any effect on the algorithms, as the signatures are not influenced by different decompositions. We summarize the procedure in Algorithm 1.

Even though $|V|$ different times are needed for distinct signatures we noticed in the experiments that in practice fewer times are actually needed.

The complexity of the algorithm is exponential when there exists an exponential number of automorphisms, for example in a clique where all matches are possible, and it can be exponential for a polynomial number of symmetries. Still, in all the experiments we performed a branch never back-folded, hence polynomial time, in the number of vertices, was measured.

It lead us to define an optimistic isomorphism algorithm. We do not perform a split in the solution space but rather choose a single path. While this process is efficient, we can not guarantee its success. Yet, we did not encounter a case in which it failed.

The spectrum can be evaluated at a complexity of $O(V^3)$, but in practice we need only partial decomposition and the power method becomes a good alternative. In our experiments we used matlab eigendecomposition functions. For large graphs we only used part of the eigenfunctions (around 1%) and received perfect results.

We use a small (constant) number of times (scales) which means that in the each stage, where one additional anchor vertex is added, it requires $O(|V|^2)$ in the worst case to find all matches between signatures. In practice, we use an approximate nearest neighbors (ANN) framework for those comparisons which is $O(|V|\log|V|)$.

In Algorithm 2 we do not perform a split, hence, given k anchor vertices the complexity is $O(k|V|\log|V|)$, where usually k is very small. In Algorithm 1 an exponential number of ANN evaluations with respect to the number of vertices can be required, but in all the graphs we examined only a linear number of ANN evaluations with respect to the number of symmetries was measured.

7 NUMERICAL VALIDATION

In the following experiments we used 10 different times spreading linearly from 10^{-1} to 10^{-4}. We found the framework robust for different times given small to medium graphs. We used all eigenvectores in the construction of the signatures. Basic shapes such as lines,

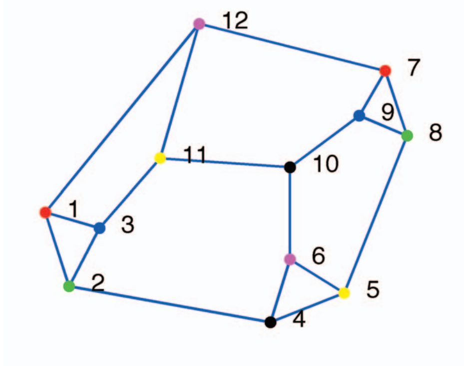

Fig. 5. The Frucht graph has no nontrivial automorphisms. No additional matchings were found by the algorithm.

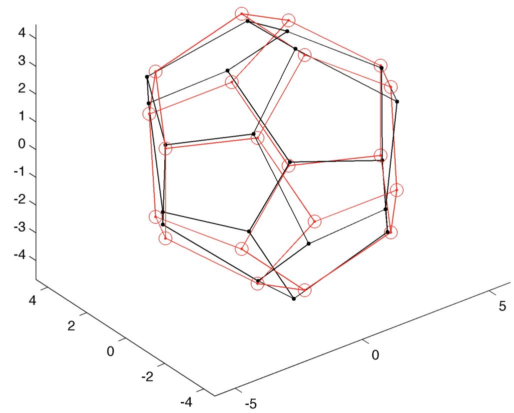

Fig. 9. A Dodecahedron before (black) and after (red) adding Gaussian noise.

triangles and squares are the first to be explored. In Figure 1 we see that all automorphisms were found. More challenging graphs are presented in Figure 2.

We applied our method on several benchmarks. Figure 3 depicts 9 out of the 336 automorphisms that were found for the Coxeter graph. It is a 3-regular graph with 28 vertices and 42 edges. Using the proposed algorithm all automorphisms were detected. Next, we considered the Dodecahedral graph which is the platonic graph corresponding to the connectivity of the vertices of a Dodecahedron. Figure 4 depicts 9 out of the 120 automorphisms. Again, all automorphisms were found. After evaluating the eigenvalues and eigenfunctions we measured linear complexity of the algorithm for both graphs. This means that, once a match between vertices was marked, the algorithm did not disqualify it in the following steps.

Next, we checked the Frucht graph shown in Figure 5, which is a 3-regular graph with 12 vertices and 18 edges but with no nontrivial symmetries. As expected, no additional matching signatures were found.

Finally, we searched for isomorphisms between two graphs. In each experiment we show two graphs and the isometry by matching colors and numbers. In Figure 6 we provide one isometry between nodes of the Coxeter graph after randomly shuffling its indices, and in a similar manner for the Dodecahedral graph in Figure 7. The last small scale experiment was done on a bipartite graph, as seen in Figure 8, where we present one of the 48 possible isometries with the connectivity table below.

We compared our results to the results obtained by using the *bliss* package, and found that for random graphs which have only one automorphism, the *bliss* package is faster. This is due to the time needed for spectral decomposition. Yet *bliss* failed to find all symmetries even for simple cases. It did find all 120 automorphisms of the Dodecahedral graph, but only 12 out of the 336 of the Coxeter graph, while the proposed method found all of them. In addition, Jiang *et. al.* [9] evaluated all the symmetries of the Dodecahedral graph on a 2.4GHz computer using their branch-and-bound approach and reported it took 131.2 seconds. Using the proposed method, we found all symmetries after 0.35 seconds, including the eigendecomposition step, on a 2.7GHz computer running matlab as well.

In Algorithms 1 and 2, we stated that two signatures are similar if they have equal values. In practice, we considered two signatures to be equal if the l_1 difference between them was extremely small (10^{-10}). In order to find approximate symmetries we use the weighted Laplacian and change the strict equality constraint to a threshold barriar. We tested our scheme on a Dodecahedron (Figure 9). Instead of using its adjacently matrix, which is the Dodecahedral graph we previously examined, we chose weights as the distances between vertices. In Table 1 we show that for a low threshold only the identity is found, but as the threshold increases we find additional symmetries. We repeated the experiment on different noisy versions of the Dodecahedron. The original length of each edge was $\left(1+\sqrt{5}\right)/2$, and we added 5, 10, 15, 20 and 25 percentage of a Gaussian noise with a zero mean and a variance of one. The barrier on the signatures' proximity was increased by a factor of 1.5 for each experiment starting with 10^{-8}.

We tested the framework on large random graphs. We found that in all cases only one (non-constant) eigenvector was actually needed to find the matches. We used a 2.7GHz computer with 4GB memory, with matlab code for all stages. We repeated the experiment 50 times on graphs with 1K to 7.5K vertices, and 10K to 750K edges, and provide the average timing of the entire process in Table 2 and the matching part alone (without eigendecomposition) in Table 3.

Finally, we tested our framework on strongly regular graphs. Such graphs are known for high number of automorphisms with various connectivities. A regular graph with v vertices and degree k is called strongly regular with parameters λ and μ if every two adjacent vertices have λ common neighbors, and every two non-adjacent vertices have μ common neighbors. We denote such a graph by (v, k, λ, μ). In this experiment we used 10 different times spreading linearly between 10^{-3} to 10^{-1}. In

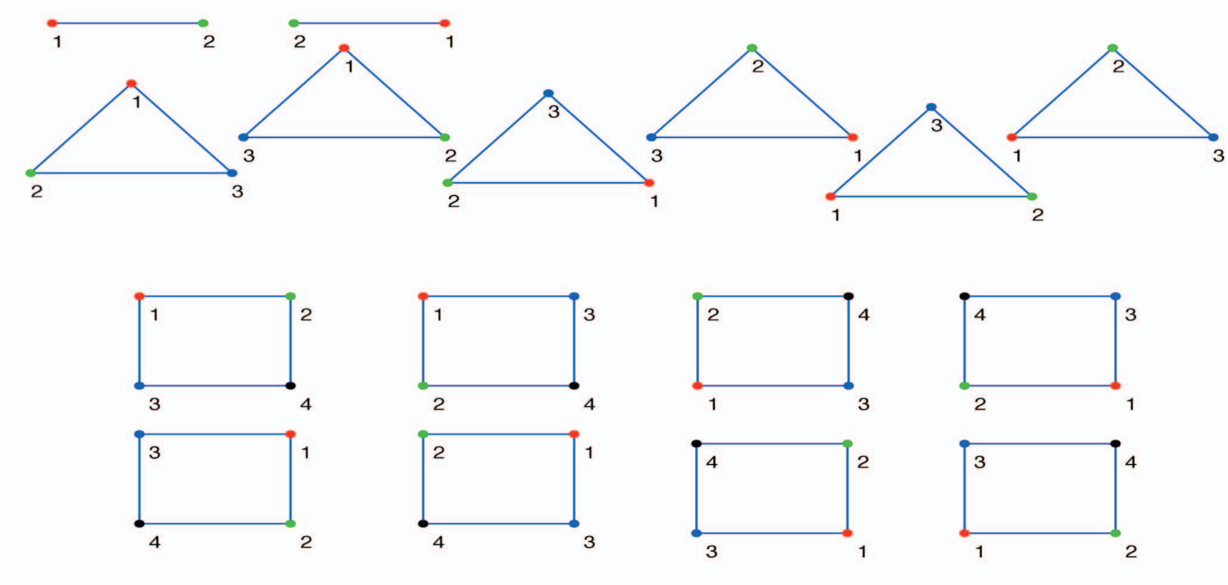

Fig. 1. Automorphisms of basic shapes. Matching vertices have similar numbers and colors.

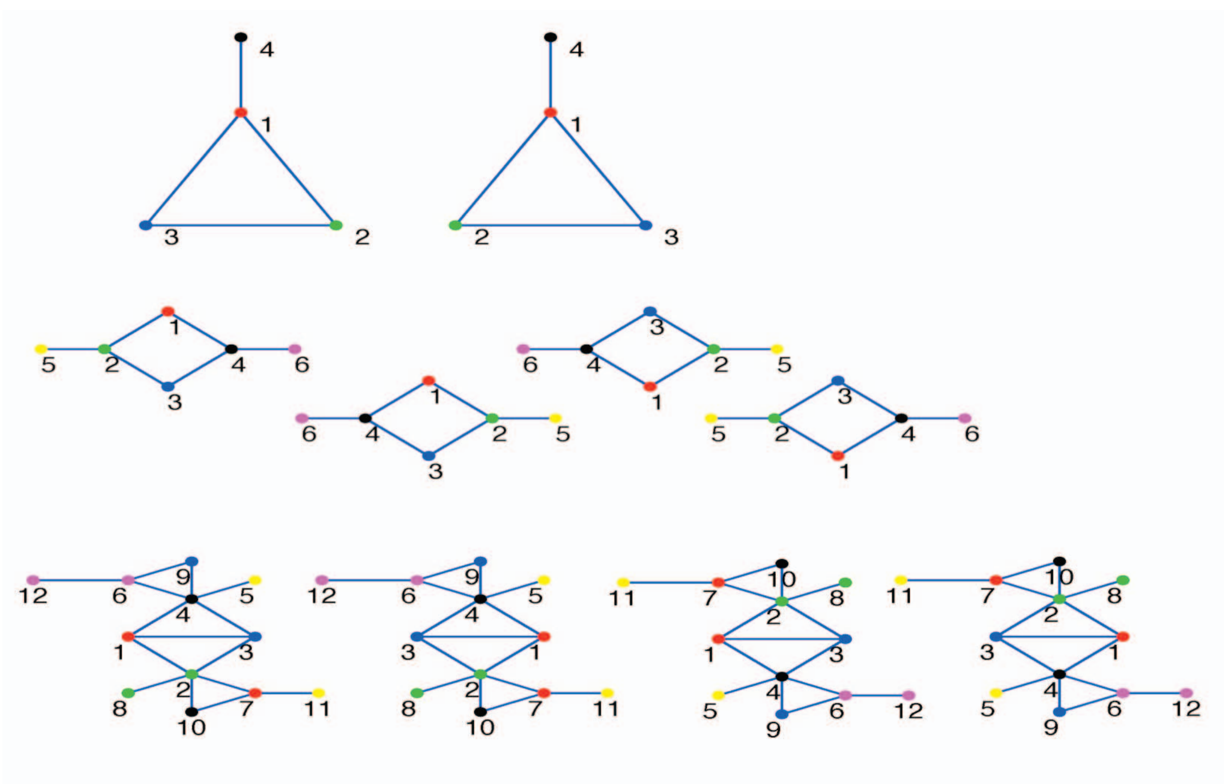

Fig. 2. Automorphisms of shapes. Matching vertices have similar numbers and colors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

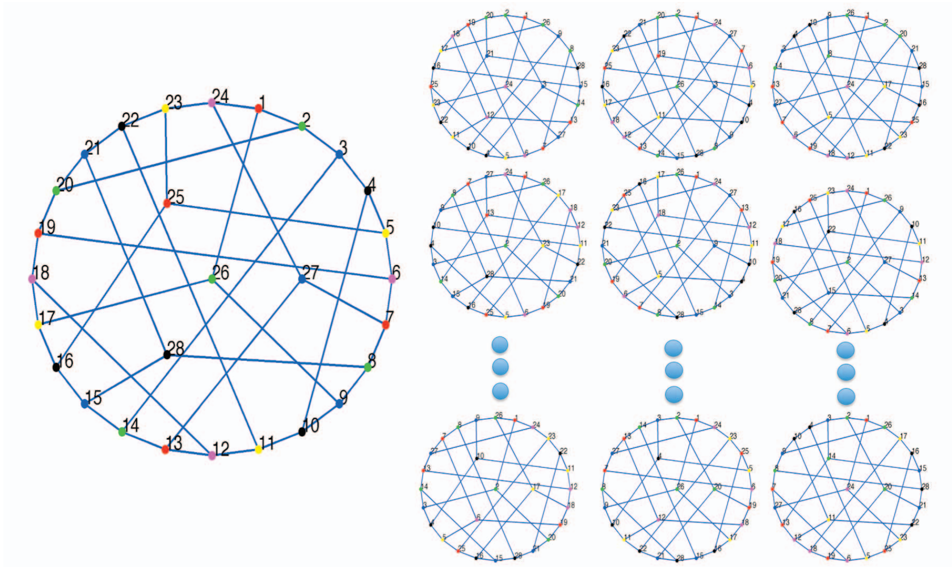

Fig. 3. 9 out of the 336 automorphisms of the Coxeter graph. All self matchings were detected.

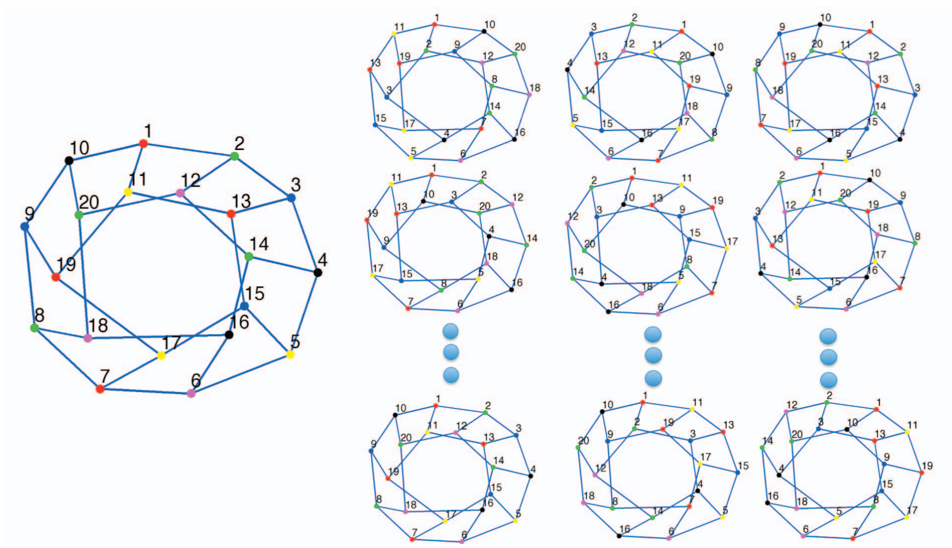

Fig. 4. 9 out of the 120 automorphisms of the Dodecohedral graph. All self matchings were detected.

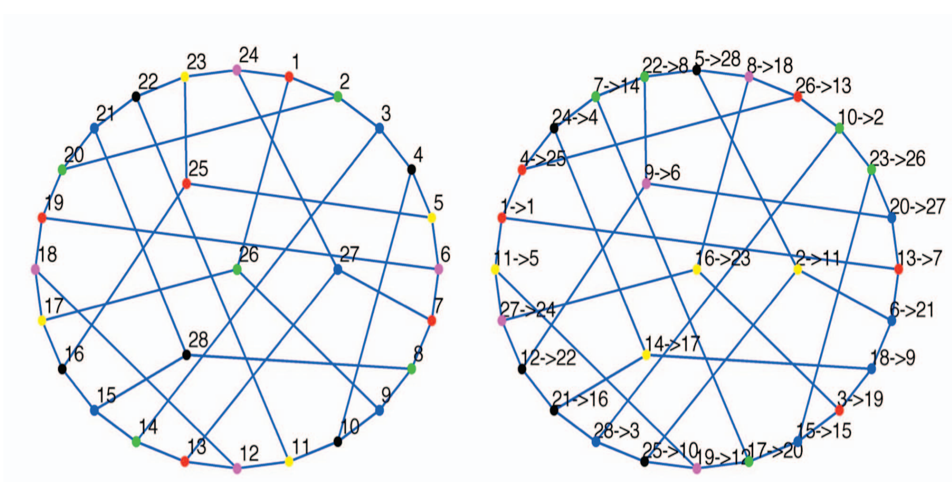

Fig. 6. One isometry between two coxeter graphs with different indexing. Similar colors and arrows represent the isometry.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

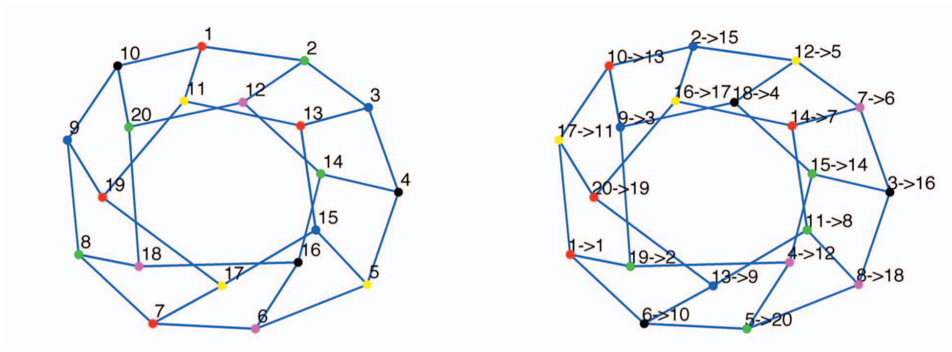

Fig. 7. One isometry between two dodecahedral graphs with different indexing. Similar colors and arrows represent the isometry.

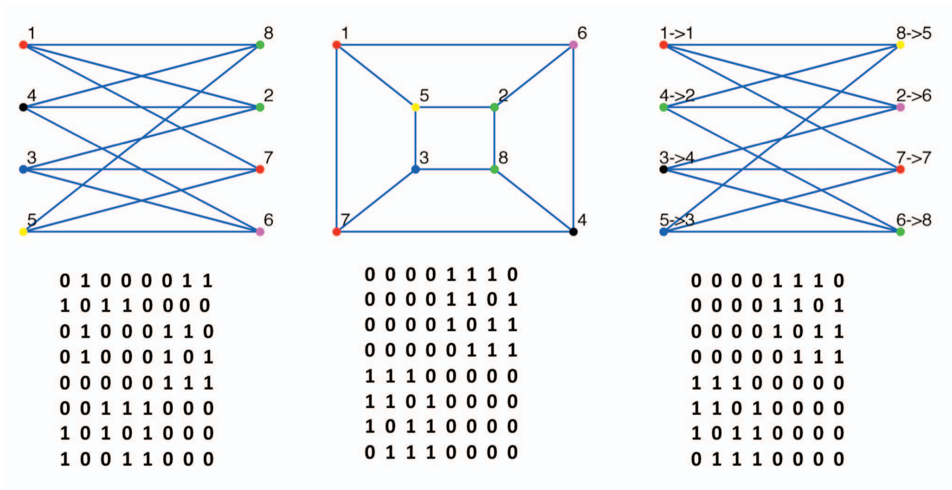

Fig. 8. One isometry between two bipartite graphs (left two columns) with different indexing. Similar colors and arrows represent the isometry (right column). We provide the connectivity tables on the bottom row.

Noise \ TH 1e-8 $\times(\frac{3}{2})^x$	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
0 %	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
5 %	1	1	1	1	1	1	3	26	104	120	120	120	120	120	120
10 %	1	1	1	1	1	1	1	1	4	16	57	119	120	120	120
15 %	1	1	1	1	1	1	1	1	1	3	24	99	120	120	120
20 %	1	1	1	1	1	1	1	1	1	1	2	11	74	120	120
25 %	1	1	1	1	1	1	1	1	1	1	2	11	44	110	120

TABLE 1
Symmetries of a noisy Dodecahedron. In each row, different magnitude of white noise was added to vertices' location. In each column we increased the threshold of signatures proximity. As the threshold increases we find more optional symmetries.

Table 4 we provide the timing results of the framework using the optimistic algorithm for isomorphisms search. For each graph shown in the left column, we provide the timing (second column) for full eigendecomposition and the timing results of five experiments (columns three to seven). We search for an isomorphisms between the strongly connected graph before and after we permute the vertices (third column). We repeated the experiment after we remove and add one and two edges, such that no isomorphism exists.

8 Conclusions

We analyzed graph automorphisms and isomorphisms from a spectral point of view, based on concatenation of heat kernel signatures. We found the scheme to be efficient, robust and feasible for practical usage. The arbitrary choice of time in the algorithm may not be sufficient for all graphs, and further research is needed especially for large challenging graphs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Vertices \ Edges	10	20	30	40	50	60	70	80	90	100
1000	0.15	0.14	0.15	0.17	0.20	0.22	0.24	0.26	0.28	0.31
1500	0.16	0.19	0.24	0.24	0.28	0.34	0.39	0.42	0.47	0.47
2000	0.19	0.31	0.37	0.44	0.49	0.54	0.62	0.61	0.66	0.72
2500	0.30	0.43	0.58	0.77	0.81	1.02	0.96	0.96	0.91	0.97
3000	0.53	0.53	0.93	1.22	1.31	1.25	1.45	1.50	1.59	1.42
3500	0.83	0.94	1.67	1.72	1.85	1.63	2.08	2.17	2.11	1.88
4000	1.50	1.32	2.31	2.06	2.43	2.93	2.50	2.72	3.03	3.10
4500	1.66	1.32	2.52	3.95	3.97	3.76	4.67	4.48	4.09	4.18
5000	3.77	2.58	4.91	5.82	5.25	5.15	5.60	5.25	6.05	5.28
5500	3.30	4.08	6.11	7.18	6.84	7.67	9.40	7.83	9.09	8.41
6000	7.77	6.56	8.90	9.22	9.28	13.51	13.11	12.10	11.24	11.88
6500	10.49	8.92	11.50	14.24	15.99	18.13	16.41	17.39	12.96	15.65
7000	12.82	11.70	13.79	19.18	22.20	26.46	22.43	21.41	21.69	22.57
7500	23.36	14.00	17.32	28.59	28.09	24.09	32.13	27.45	26.12	26.50

TABLE 2

Timing (seconds) of random graphs isomorphism search. The number of vertices increases in each row, and the number of edges per vertex increases per column. The largest graph has 7.5K vertices, and 0.75M edges.

Vertices \ Edges	10	20	30	40	50	60	70	80	90	100
1000	0.12	0.10	0.11	0.13	0.15	0.17	0.20	0.21	0.23	0.25
1500	0.11	0.13	0.16	0.18	0.20	0.23	0.31	0.32	0.37	0.38
2000	0.15	0.18	0.21	0.25	0.29	0.34	0.40	0.43	0.47	0.50
2500	0.18	0.23	0.28	0.35	0.40	0.44	0.49	0.56	0.59	0.65
3000	0.22	0.28	0.36	0.43	0.49	0.54	0.62	0.66	0.76	0.82
3500	0.26	0.34	0.42	0.52	0.61	0.67	0.72	0.84	0.91	0.97
4000	0.31	0.40	0.51	0.62	0.73	0.82	0.91	0.99	1.02	1.11
4500	0.41	0.51	0.64	0.73	0.80	0.97	1.06	1.10	1.24	1.32
5000	0.47	0.60	0.71	0.86	0.93	1.10	1.15	1.31	1.39	1.47
5500	0.53	0.70	0.82	0.95	1.12	1.25	1.32	1.50	1.58	1.71
6000	0.63	0.77	0.97	1.11	1.24	1.40	1.63	1.68	1.86	1.96
6500	0.72	0.92	1.09	1.21	1.41	1.55	1.63	1.81	2.03	2.15
7000	0.84	1.01	1.20	1.35	1.56	1.67	1.90	2.02	2.20	2.39
7500	0.90	1.11	1.34	1.56	1.73	1.88	2.02	2.36	2.46	2.54

TABLE 3

Timing (seconds) of random graphs matching search. Similar experiment as shown in Table 2, but without the eigendecomposition preprocessing.

Acknowledgments

This research was partially supported by the Israel Science Foundation grant no. 1551/09 and 1031/12 and by Initial Training Network of the European Commission PITN-GA-2009-238702.

References

[1] J. Murrell, S. Kettle, and J. M. Tedder, *The chemical bond*. Wiley, 1988.

[2] L. Babai, W. M. Kantor, and E. M. Luks, "Computational complexity and the classification of finite simple groups," in *Symposium on Foundations of Computer Science (FOCS)*, 1983, pp. 162–171.

[3] E. M. Lukas, "Isomorphism of graphs of bounded valence can be tested in polynomial time," *Journal of Computer and System Sciences*, vol. 25, pp. 42–65, 1982.

[4] L. Babai and P. Codenotti, "Isomorphism of hypergraphs of low rank in moderately exponential time," in *Symposium on Foundations of Computer Science (FOCS)*, 2008, pp. 667–676.

[5] G. S. Lueker and S. K. Booth, "A linear time algorithm for deciding interval graph isomorphism," *Journal of the ACM (JACM)*, vol. 26, no. 2, 1979.

[6] J. E. Hopcroft and J. K. wong, "Linear time algorithm for isomorphism of planar graphs," in *ACM symposium on Theory of computing (STOC)*, 1974, pp. 172–184.

[7] L. Babai, D. Y. Grigoryev, and D. M. Mount, "Isomorphism of graphs with bounded eigenvalue multiplicity," in *ACM symposium on Theory of computing (STOC)*, 1982, pp. 310–324.

[8] M. You and A. K. C. Wong, "An algorithm for graph optimal isomorphism," in *International Conference of Pattern Recognition (ICPR)*, 1984, pp. 316–319.

[9] X. Jiang, J. Sun, and L. Guibas, "A Fourier-Theoretic Approach for Inferring Symmetries," in *Canadian Conference on Computational Geometry (CCCG)*, 2011.

[10] M. Gori, M. Maggini, and L. Sarti, "Exact and approximate graph matching using random walks," *IEEE Trans. Pattern Anal. Mach. Intell (PAMI)*, vol. 27, no. 7, pp. 1100–1111, July 2005.

[11] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell (PAMI)*, vol. 10, no. 5, September 1988.

[12] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *Association for computing machinery*, vol. 23, no. 1, 1976.

[13] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "Performance Evaluation of the VF Graph Matching Algorithm," in *International Conference on Image Analysis and Processing (ICIAP)*, 1999.

[14] ——, "A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs," *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 10, pp. 1367–1372, 2004.

[15] T. Junttila and P. Kaski, "Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs," in *Proceedings of the Workshop on Algorithm Engineering and Workshop on algorithm engineering and experiments (ALENEX)*, 2007.

[16] B. McKay, "Practical graph isomorphism," in *10th. Manitoba Conference on Numerical Mathematics and Computing*, 1981, pp. 45–87.

[17] P. T. Darga, K. A. Sakallah, and I. L. Markov, "Faster symmetry discovery using sparsity of symmetries," in *Proceedings of the 45th annual Design Automation Conference*, 2008, pp. 149–154.

[18] P. Bérard, G. Besson, and S. Gallot, "Embedding Riemannian

Graph \ Timing	Eigendecomposition	Isomorphisms	Remove 1	Remove 2	Add 1	Add 2
$(9, 4, 1, 2)$	0.000335	0.106636	0.000710	0.000576	0.000512	0.000691
$(10, 3, 0, 1)$	0.000310	0.156075	0.000722	0.000575	0.000511	0.000731
$(13, 6, 2, 3)$	0.000342	0.263750	0.000730	0.000589	0.000570	0.000775
$(15, 6, 1, 3)$	0.000314	0.269383	0.000722	0.000593	0.000543	0.000820
$(21, 10, 3, 6)$	0.000397	0.565614	0.000788	0.000590	0.000706	0.000679
$(25, 8, 3, 2)$	0.000430	0.880930	0.000777	0.000581	0.000834	0.000602
$(27, 10, 1, 5)$	0.000430	0.809530	0.000794	0.000612	0.000689	0.000707

TABLE 4

Timing (seconds) of strongly regular graphs matching search. For each graph shown in the left column, we provide the timing (second column) for full eigendecomposition and the timing results of five experiments. We search for an isomorphisms between before and after we permute the vertices (third column). We repeated the experiment after we remove and add one and two edges, such that no isomorphism exists.

manifolds by their heat kernel," *Geometric and Functional Analysis*, vol. 4, no. 4, pp. 373–398, 1994.
[19] R. M. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Proc. Symposium on Geometry Processing (SGP)*, 2007, pp. 225–233.
[20] A. Sharma, R. P. Horaud, and D. Mateus, "3D Shape Registration Using Spectral Graph Embedding and Probabilistic Matching," pp. 441–474, 2012.
[21] J. Sun, M. Ovsjanikov, and L. J. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Proc. Symposium on Geometry Processing (SGP)*, 2009.
[22] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. J. Guibas, "One point isometric matching with the heat kernel," *Proc. Symposium on Geometry Processing (SGP)*, vol. 29, no. 5, pp. 1555–1564, Jul 2010.
[23] B. Xiao, R. Wilson, and E. Hancock, "Graph characteristics from the heat kernel trace," *Pattern Recognition*, vol. 42, no. 11, pp. 2589–2606, 2009.
[24] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Full and Partial Symmetries of non-Rigid Shapes," *International Journal of Computer Vision (IJCV)*, 2010.
[25] F. R. K. Chung, *Spectral graph theory*. AMS, 1997.

Dan Raviv is a PhD candidate at the Technion, Israel institute of Technology working at the GIP-Lab. He has a bachelor degree, Summa cum Laude, in both Mathematics and Computer Science, being an alumnus of Technion's Excellence Program. After receiving his masters degree he spent a summer internship working at HP-Labs Israel. Dan's research was granted several excellence prizes including the Gutwirth and Intel awards. Last year he was elected as best lecturer and awarded the department's teaching trophy for his excellence in educating and coaching undergraduates.

Ron Kimmel is a Professor of Computer Science at the Technion where he holds the Montreal Chair in Sciences. He held a post-doctoral position at UC Berkeley and a visiting professorship at Stanford University. He has worked in various areas of image and shape analysis in computer vision, image processing, and computer graphics. Kimmel's interest in recent years has been non-rigid shape processing and analysis, medical imaging and computational biometry, numerical optimization of problems with a geometric flavor, and applications of metric geometry and differential geometry. He is an IEEE Fellow for his contributions to image processing and non-rigid shape analysis. He is an author of two books, an editor of more than a couple, and an author of numerous articles. He is the founder of the Geometric Image Processing Lab as well several successful shape acquisition, processing and analysis companies.

Alfred M. Bruckstein received his BSc and MSc degrees at the Technion, Haifa, in 1976 and 1980, respectively then earned a Ph.D. degree in Electrical Engineering at Stanford University, California. From October 1984 he has been with the Technion, where he holds of the Ollendorff Chair in Science. He served as the Dean of Technion's Graduate School (2002-2006) and as the head of Technion's Excellence Program for undergraduates (2007-2012). He held visiting positions at Stanford University, Groningen University, M.I.T, Bell Labs, Tsinghua University and is presently also a visiting professor at Nanyang Technological University in Singapore. His research interests are in swarm/ant robotics, image and signal processing, analysis and synthesis, pattern recognition, and various aspects of applied geometry. Professor Bruckstein authored and co-authored about one hundred and fifty journal papers in the fields of interest mentioned. He is a member of SIAM, the American Mathematical Society and the Mathematical Association of America.