

Compact Form of the Pseudo–inverse Matrix in the Approximation of a Star Graph Using the Conductance Electrical Model (CEM) *

Manuel Igelmo¹ and Alberto Sanfeliu^{1,2}

¹ Universitat Politècnica de Catalunya (UPC)

² Institut de Robòtica i Informàtica Industrial (UPC–CSIC)

migelmo@xtec.cat, sanfeliu@iri.upc.edu

<http://www.upc.edu>, <http://www.iri.upc.edu>

Abstract. The Conductance Electrical Model (CEM) transforms a graph into a circuit and can be use to do "inexact graph isomorphism" as it was shown in [13]. In second stage of this process, we transform the circuit r_{eq} in a star circuit, using the Moore–Penrose pseudo–inverse of a matrix for which there is a general formula that requires transpose, multiply and invert matrices with a time complexity of $O(N^4)$, where N is the number of nodes of the graph. However, due to the special structure of the star transformation, we are able to exploit this special structure to compute the pseudo–inverse without using the general Moore–Penrose formula. We have developed a closed formula that can compute the elements of the pseudo–inverse without using that formula, that means without multiplying matrices neither doing the matrix inversion and that moreover can be computed in $O(N^3)$. This method also eliminates the problems due to computer rounding and due to bad–conditioned problems in mathematical terms.

1 Introduction

Graphs have been successfully applied in various fields such as chemistry and biochemistry; transportation, telephony and computers networks, speech recognition and computer vision [1]. Examples of graphs in computer vision can be seen in [7] and they usually have a large number of nodes and/or edges.

The methods for graph and sub-graph matching are based on enumerative techniques [2,3], edit operations [4,5,6], spectral methods [8], expectation-maximization [9], random walks [10], genetics algorithms [11] and probabilistic approximations [12]. The time complexity in the enumerative and edit operation methods is NP–complete while in the other inexact methods it is polynomially bounded. Only in the enumerative solutions there exist an exact solution, in the other cases only graph matching approximations can be obtained.

* This research was conducted at the Institut de Robòtica i Informàtica Industrial (CSIC-UPC). It was partially supported by the CICYT project RobTaskCoop (DPI2010-17112) and the MIPRCV Ingenio Consolider 2010 (CSD2007-018)

In [13], we proposed a model to replace a graph by a circuit and we use the methods of Circuit Theory to solve the graph isomorphism. In order to compare two graphs, the method transform a graph G into a star circuit G^* using the following stages (see article [13]):

First Stage: Computation of the equivalent circuit resistances r_{eq}^G

1. Obtain the adjacency matrix, A^G
2. Compute the Laplacian matrix, Y^G obtained from A^G
3. Eliminate one row and one column; the one that belong to the node that will be consider the reference node in the electrical circuit (any node can be consider the reference node) and obtain the new matrix X^G
4. Apply the Ohm Law and compute the r_{eq}^G

Second Stage: Computation of the star circuit G^*

5. Obtain the branch resistances of the star circuit \mathbf{r} , by computing $\mathbf{r} = B^+ \mathbf{r}_{eq} = (B^t B)^{-1} B^t \mathbf{r}_{eq}$

In the first stage, each undirected weighted graph, G , of N nodes is transformed in the circuit CEM model, a passive resistive circuit where the weights of the edges are the conductance in siemens, and then in the adjacency matrix A^G . In the second stage, the CEM model is transformed in a star circuit, G^* , with $N + 1$ nodes and N branches by minimizing the mean square error of the $N(N - 1)/2$ equivalent resistances. This operation requires the calculation of the Moore–Penrose pseudo–inverse (hereinafter simply pseudo–inverse) of the B matrix by the general formula involving the product and inversion of matrices. In this work we have developed a new close form that computes the pseudo–inverse of B without the need of matrix multiplication and inversion.

2 Star Approximation Using CEM

By using the first described stage, we obtain the $N(N - 1)/2$ equivalent resistances ($r_{eq_{ij}}$) which values can be represented by a column vector

$$\mathbf{r}_{eq} = (r_{eq_{1,2}}, r_{eq_{1,3}}, \dots, r_{eq_{1,N}}, r_{eq_{2,3}}, \dots, r_{eq_{2,N}}, \dots, r_{eq_{N-3,N-1}}, r_{eq_{N-2,N-1}}, r_{eq_{N-1,N}})^t$$

In the work [13] we proposed to approximate the original circuit by a star circuit (with N branches and $N + 1$ nodes including one node in the center of the star) with one resistance (r_i) for each branch. These values can be written as a column vector

$$\mathbf{r} = (r_1, r_2, \dots, r_N)^t$$

Also there are $N(N - 1)/2$ equivalent resistances in the star circuit. Note the central node is not involved in the calculation of the equivalent resistances. The equivalent resistances can be written as a column vector

$$\mathbf{r}'_{eq} = (r'_{eq_{1,2}}, r'_{eq_{1,3}}, \dots, r'_{eq_{1,N}}, r'_{eq_{2,3}}, \dots, r'_{eq_{2,N}}, \dots, r'_{eq_{N-3,N-1}}, r'_{eq_{N-2,N-1}}, r'_{eq_{N-1,N}})^t$$

It is easy to see that $r'_{eq_{ij}} = r_i + r_j$, since the equivalent resistance between two nodes in the star circuit, is the association of two serial resistances. Then we have $\mathbf{r}'_{eq} = B\mathbf{r}$ where B is the matrix shown in (1)

$$B = \left(\begin{array}{cccc}
 1 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 \cdots 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 1 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 1 \\
 \hline
 0 & 1 & 1 & 0 \cdots 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 \cdots 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 1 & 0 & 0 \cdots 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 1 \\
 \hline
 0 & 0 & 1 & 1 \cdots 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 1 & 0 \cdots 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 \cdots 0 & 0 & 1 & 0 \\
 \hline
 0 & 0 & 0 & 1 \cdots 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & 1 \cdots 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 \cdots 0 & 0 & 1 & 0 \\
 \hline
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & 0 \cdots 0 & 1 & 1 & 1
 \end{array} \right) \quad \left. \begin{array}{l}
 \} N - 1 \text{ rows} \\
 \} N - 2 \text{ rows} \\
 \} N - 3 \text{ rows} \\
 \} N - 4 \text{ rows} \\
 \} 1 \text{ row}
 \end{array} \right. \quad (1)$$

The approximation discussed above must be understood as the search for the values of \mathbf{r} such that \mathbf{r}'_{eq} is approximately equal to \mathbf{r}_{eq} , in the sense of minimizing the mean square error between \mathbf{r}_{eq} and \mathbf{r}'_{eq} is given by

$$\mathbf{r} = (B^t B)^{-1} B^t \mathbf{r}_{eq}$$

where

$$B^+ = (B^t B)^{-1} B^t \quad (2)$$

is known as the pseudo-inverse of B , note that B^+ has N rows and $N(N - 1)/2$ columns. The above equation we can finally be written

$$\mathbf{r} = B^+ \mathbf{r}_{eq}$$

3 Compact Form of the Pseudo-inverse

To obtain the pseudo-inverse (B^+) of any matrix B by the (2) expression it is necessary to make a matrix inversion, two products of matrices and matrix transpose. But for the particular case that the matrix B has the form given in (1), it is not necessary to use the (2) expression. This substantially simplifies calculations as discussed in the following theorem.

Theorem 1. *Let B be the matrix with the structure shown in (1) with $N \neq 1$ and $N \neq 2$, then its pseudo-inverse is*

$$B^+ = \frac{1}{(N-1)(N-2)} [(N-1)B^t - \mathbf{1}_{N,N(N-1)/2}] \tag{3}$$

where N is the number of columns of the matrix B and $\mathbf{1}_{N,N(N-1)/2}$ is a matrix full of ones with N rows and $N(N-1)/2$ columns.

Proof. We call M the result of $B^t B$ then it is easy to see that

$$M = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} N-1 & 1 & 1 & \dots & 1 \\ 1 & N-1 & 1 & \dots & 1 \\ 1 & 1 & N-1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & N-1 \end{pmatrix} \tag{4}$$

where M is a square matrix of order N . To calculate M^{-1} we will use that

$$M^{-1} = \frac{M^*}{|M|}$$

where M^* is the adjugate matrix and $|M|$ is the determinant that must necessarily be non-zero so that the inverse matrix does exist. Applying the formula (8) of the Theorem 2 obtained in the Annex and substituting n by N and k by $N - 1$ then

$$|M| = 2(N - 1)(N - 2)^{N-1} \tag{5}$$

Performing the same substitutions in (9) of the Theorem 3 for the adjugate matrix we obtain the following

$$M^* = (N - 2)^{N-2} \begin{pmatrix} 2N - 3 & -1 & -1 & \cdots & -1 \\ -1 & 2N - 3 & -1 & \cdots & -1 \\ -1 & -1 & 2N - 3 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 2N - 3 \end{pmatrix} \tag{6}$$

Dividing the expressions (5) and (6) we obtain

$$M^{-1} = \frac{1}{2(N - 1)(N - 2)} \begin{pmatrix} 2N - 3 & -1 & -1 & \cdots & -1 \\ -1 & 2N - 3 & -1 & \cdots & -1 \\ -1 & -1 & 2N - 3 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 2N - 3 \end{pmatrix}$$

Note that M^{-1} can be written as

$$M^{-1} = \frac{1}{2(N - 1)(N - 2)} [2(N - 1)\mathbb{I}_N - \mathbb{1}_{N,N}]$$

where \mathbb{I}_N is the identity matrix of order N . We finally have

$$\begin{aligned} B^+ = M^{-1}B^t &= \frac{1}{2(N - 1)(N - 2)} [2(N - 1)\mathbb{I}_N - \mathbb{1}_{N,N}]B^t = \\ &= \frac{1}{2(N - 1)(N - 2)} [2(N - 1)\mathbb{I}_N B^t - \mathbb{1}_{N,N}B^t] = \\ &= \frac{1}{2(N - 1)(N - 2)} [2(N - 1)B^t - 2\mathbb{1}_{N,N(N-1)/2}] = \\ &= \frac{1}{(N - 1)(N - 2)} [(N - 1)B^t - \mathbb{1}_{N,N(N-1)/2}] \end{aligned}$$

The last step is due to the fact that all the columns of B^t add the constant 2. \square

4 Conclusions and Advantages of the Compact Form of the Pseudo-inverse

We have shown in this article that there is a way of computing the pseudo-inverse of the second stage of the inexact isomorphism computation, without requiring matrix transpose, inversion and the multiplication of matrices, because we can build the pseudo-inverse in a direct way.

The advantages of the calculation of pseudo-inverse by the compact formula (3) versus the general formula (2) are:

- 1) The computational complexity is reduced from $O(N^4)$ to $O(N^3)$.
- 2) This improvement avoids the problem of numerical resolution in matrix pseudo-inversion on a computer (numerical stability is ensured) and also avoids potential bad-conditioned problems in mathematical terms.
- 3) The elements of matrix B^+ may be obtained on the fly. It is not necessary to work in memory with the entire matrix, therefore this improvement is important in systems with low memory.

5 Annex

Let be Q a matrix of order n

$$Q = \begin{pmatrix} k & 1 & 1 & \cdots & 1 \\ 1 & k & 1 & \cdots & 1 \\ 1 & 1 & k & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & k \end{pmatrix} \tag{7}$$

The following two theorems are fulfilled:

Theorem 2. *The determinant of the matrix Q is*

$$|Q| = (k + n - 1)(k - 1)^{n-1} \tag{8}$$

Proof. Let us going to obtain the upper triangular matrix. For each row it has to be added all the columns to first column

$$|Q| = \begin{vmatrix} k+n-1 & 1 & 1 & \cdots & 1 \\ k+n-1 & k & 1 & \cdots & 1 \\ k+n-1 & 1 & k & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k+n-1 & 1 & 1 & \cdots & k \end{vmatrix} \quad \text{then} \quad |Q| = (k+n-1) \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & k & 1 & \cdots & 1 \\ 1 & 1 & k & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & k \end{vmatrix}$$

Each row is replaced, except the first row that is obtained by subtracting the first row

$$|Q| = (k+n-1) \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & k-1 & 0 & \cdots & 0 \\ 0 & 0 & k-1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & k-1 \end{vmatrix}$$

Since the determinant of a triangular matrix is the product of the diagonal elements, then

$$|Q| = (k + n - 1)(k - 1)^{n-1} \quad \square$$

Corollary 1. *The determinant of Q is not zero if and only if $k \neq 1$ and $k \neq 1 - n$.*

Theorem 3. *The adjoint matrix (Q^*) of the matrix Q is*

$$Q^* = (k - 1)^{n-2} \begin{pmatrix} k + n - 2 & -1 & -1 & \cdots & -1 \\ -1 & k + n - 2 & -1 & \cdots & -1 \\ -1 & -1 & k + n - 2 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & k + n - 2 \end{pmatrix} \quad (9)$$

Proof. To proof this theorem we have to divide the problem in two parts: (i) the calculation of the diagonal adjoints (Q_{ii}) and (ii) the calculation of the off-diagonal adjoints (Q_{ij}).

(i) Calculation of the Diagonal Adjoints (Q_{ii}).

The adjoint of a diagonal element (all the adjoints of the diagonal elements are identical) will be a determinant of order $n - 1$. Applying the formula (8) of Theorem 2 we obtain the following expression

$$Q_{ii} = \begin{vmatrix} k & 1 & 1 & \cdots & 1 \\ 1 & k & 1 & \cdots & 1 \\ 1 & 1 & k & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & k \end{vmatrix} = (k + n - 2) \cdot (k - 1)^{n-2}$$

(ii) Calculation of the Off-Diagonal Adjoints (Q_{ij})

As it was shown in (10) to calculate the adjoint Q_{ij} ($i \neq j$) it must be removed the row i and column j (solid line) of Q .

$$Q = \left(\begin{array}{cccccc|c|cccccc} k & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 & & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \\ 1 & \cdots & k & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & \\ \hline 1 & \cdots & 1 & k & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & \\ \hline 1 & \cdots & 1 & 1 & k & \cdots & 1 & 1 & 1 & \cdots & 1 & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \\ 1 & \cdots & 1 & 1 & 1 & \cdots & k & 1 & 1 & \cdots & 1 & \\ 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & k & 1 & \cdots & 1 & \\ 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & k & \cdots & 1 & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \\ 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & k & \end{array} \right) \quad (10)$$

Thereafter, it easy to see in (11) that only appears one row and one column with all elements with value one (solid line) in the adjoint of Q_{ij} (with $i \neq j$)

$$C_{ij} = (-1)^{i+j} \begin{vmatrix} k \cdots 1 & 1 & 1 \cdots 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 \cdots k & 1 & 1 \cdots 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 \cdots 1 & 1 & k \cdots 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 \cdots 1 & 1 & 1 \cdots k & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \hline 1 \cdots 1 & 1 & 1 \cdots 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \hline 1 \cdots 1 & 1 & 1 \cdots 1 & k \cdots 1 & \cdots & 1 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 \cdots 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & k \end{vmatrix} \tag{11}$$

This row will be permuted to the first row. We will proceed in a similar way for the column. For each permutation the determinant changes its sign.

Suppose that $i < j$ then the column filled with ones will appear at position i meanwhile the row with all ones will appear at position $j - 1$. Therefore the number of permutations (and consequent changes of sign) of the row and column with all ones is $j - 2$ and $i - 1$ respectively, and the determinant is affected by $(-1)^{i+j-3}$. Analogous results are obtained assuming $j < i$.

Summarizing, the coefficient that multiplies the determinant is $(-1)^{i+j}(-1)^{i+j-3}$. It will always have the value -1 , because the exponent is always odd, as it can be seen in

$$(-1)^{i+j}(-1)^{i+j-3} = (-1)^{2i+2j-3} = (-1)^{2(i+j)-3} = -1$$

Then the adjoint is as follows ($i \neq j$)

$$Q_{ij} = - \begin{vmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & k & 1 & 1 & \cdots & 1 \\ 1 & 1 & k & 1 & \cdots & 1 \\ 1 & 1 & 1 & k & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & \cdots & k \end{vmatrix}$$

The calculation of this determinant is similar to that of Theorem 2, for $i \neq j$.

$$C_{ij} = -(k - 1)^{n-2}$$

Finally we will have

$$Q^* = (k - 1)^{n-2} \begin{pmatrix} k + n - 2 & -1 & -1 & \cdots & -1 \\ -1 & k + n - 2 & -1 & \cdots & -1 \\ -1 & -1 & k + n - 2 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & k + n - 2 \end{pmatrix} \quad \square$$

References

1. Bunke, H., Sanfeliu, A.: Syntactic and Structural Pattern Recognition – Theory and Applications. Series in Computer Science, vol. 7. World Scientific Publishing Co. Pte. Ltd, Singapore (1990)
2. Ullman, J.R.: An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery* 23(1), 31–42 (1976)
3. Messmer, B.T., Bunke, H.: A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(5), 493–504 (1998)
4. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Syst. Man Cybern. SMC-13*(3), 353–362 (1983)
5. Neuhaus, M., Bunke, H.: Automatic learning of cost function for graph edit distance. *Information Sciences* 177, 239–247 (2007)
6. Neuhaus, M., Bunke, H.: Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition* 39(10), 1852–1863 (2006)
7. Sanfeliu, A., Alquézar, R., Andrade, J., Climent, J., Serratosa, F., Vergés, J.: Graph-based representations and techniques for image processing and image analysis. *Pattern Recognition* 35, 639–650 (2002)
8. Umeyama, S.: An Eigendecomposition approach to weighted graph matching problems. *IEEE Trans. PAMI* 10, 695–703 (1998)
9. Robles-Kelly, A., Hancock, E.R.: An expectation-maximisation framework for segmentation and grouping. *Image and Vision Computing* 20, 725–738 (2002)
10. Gori, M., Maggini, M., Sarti, L.: Exact and Approximate Graph Matching Using Random Walks. *Pattern Anal. and Mach. Intelligence* 27(7), 1100–1111 (2005)
11. Cross, A.D.J., Wilson, R.C., Hancock, E.R.: Inexact Graph Matching Using Genetic Search. *Pattern Recognition* 30, 953–970 (1997)
12. Wilson, R.C., Hancock, E.R.: Structural matching by discrete relaxation. *Pattern Analysis and Machine Intelligence* 19(6), 634–648 (1997)
13. Igelmo, M., Sanfeliu, A., Ferrer, M.: A Conductance Electrical Model for Representing and Matching Weighted Undirected Graphs. In: *Proceedings of the International Conference on Pattern Recognition (ICPR 2010)*, pp. 958–961 (2010)