Proceedings of the Fifth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 13–15, 2008, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-717-8 / CD: 978-0-88986-718-5

TOPOLOGICAL AND DIRECTIONAL LOGO LAYOUT INDEXING USING HERMITIAN SPECTRA

Reinier H. van Leuken
Information and Computing Sciences
Universiteit Utrecht, The Netherlands
reinier@cs.uu.nl

Olga Symonova
Fondazione Graphitech
Trento, Italy
olga.symonova@graphitech.it

Remco C. Veltkamp
Information and Computing Sciences
Universiteit Utrecht, The Netherlands
remco.veltkamp@cs.uu.nl

ABSTRACT
To evaluate similarity between two images, the layout or configuration of the shapes is an important feature besides geometrical shape similarity. In particular, trademark image retrieval is an application domain where layout similarity is important, and in many cases overlooked. In this paper, we present a graph-based encoding of layout, in which both directional and topological layout information is stored. A Hermitian matrix is associated to each graph, and contains all the information that is present in the graph. The spectra of these Hermitian matrices are used for indexing purposes. By obeying several constraints on the construction of the Hermitian matrices, we can mimic the spectral behaviour of Laplacian matrices, which are proven to be successful representations in retrieval environments. Experiments show the improved representational power of the proposed approach over spectral methods using Laplacian matrices.

KEY WORDS
Indexing, image retrieval, trademarks, Laplacian, Hermitian, spectra

1. Introduction

The key function of any indexing algorithm is to speed up content-based retrieval of objects or models that are stored in a database, by selecting a small set of candidate objects that are either presented to the user, or passed on to a more refined matching unit in the retrieval pipeline. At this matching level, more accurate and more expensive matching algorithms can be deployed because of the reduced size of the set of objects that is under inspection. At the indexing level however, comparison of objects should be efficient and it must be possible to prune the database, i.e. the database must be partitioned in such a way that similar models are positioned close to each other. Only then objects that are far from the query object can be discarded without further inspection.

Naturally, the representation of the objects in the index and the accuracy and efficiency with which non-similar objects can be discarded are closely related. The objects that are under investigation in this work are logo and trademark images, or any kind of image in general where the layout of the individual image components (as opposed to their shape characteristics) is important for similarity evaluation [10]. In content-based trademark image retrieval, layout can play a large role in identifying trademark infringement. See for an example Figure 1, where the configuration of the individual shapes is one of the most important properties. Suppose that in all three cases the five circles are returned as a result of image segmentation (which would be the ideal segmentation), it is impossible to distinguish between the images without any notion of layout in the representation. In this case, indexing algorithms (without layout information) will be less efficient because the set of candidate models will be unnecessary large. More importantly, indexing algorithms can be less accurate by ignoring layout. See Figure 2 for an illustration of a case where a low similarity score will be calculated for similar images, if only shape similarity is taken into account. If one of these images is a query, neither of the other two will be returned based on shape similarity. However, according to trademark experts, if these image were to be registered as real trademarks within similar product or service categories, a conflict of uniqueness may arise [10].

Within the area of content-based image retrieval, a lot of work has been devoted to spatially oriented retrieval. One of the most popular techniques often used for this purpose is based on string matching. To produce the strings that encode layout, the centres of mass of all objects are projected on the x and y axes. By taking objects from left to right and from below to above, and by representing these objects by a class identifier, two one-dimensional strings are formed that together form the 2D-String [3]. A number of modifications and extensions to this idea have been presented, see [9, 6, 2] for a some examples. A major drawback of these symbolic projection methods is that in general they are not rotation invariant.

In this paper, we propose a new spectral encoding for layout of shapes that can be represented and compared efficiently. Recent studies [11] have shown how spectral representations of layout can be used to index trademark collections. With the proposed encoding however, that follows some of the ideas of [12], we are able to discriminate better between different configurations, as we take into account more information without sacrificing any efficiency.

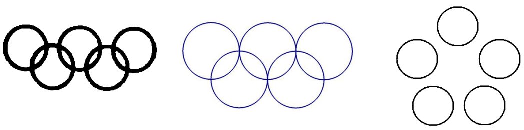


Figure 1. Example of different configurations of the same primitive shapes, with decreasing layout similarity from lefttoright.

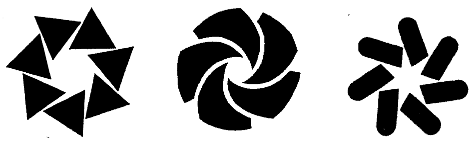


Figure 2. Three trademarks with similar layouts, but dissimilar primitive shapes.

1.1 Our contributions

The main contribution of this paper is a new method for efficient retrieval of trademark images, or images in general, that is based on the layout of the different shapes the image is composed of.

To this end, a graph is constructed for each image in which the layout of the trademark is encoded. After associating a matrix with each graph, the spectra (sorted sets of eigenvalues) of these matrices are compared for similarity evaluation. However, unlike most spectral methods, that usually focus on connectivity, several types of additional information are taken into account as well. For this purpose, we will use the spectrum of a Hermitian matrix. In Section 2 details about Hermitian spectra and how to match them are given.

This work proposes a way to encode both precise directional and topological relations between the components. These additional (graph) properties are reflected in the spectrum that is used for similarity evaluation during indexing. Details on the graph construction and calculation of attributes are given in Section 3. By obeying several constraints on the definition of the graph's topology and geometry measurements, and by encoding these values in a Hermitian matrix, we can mimic the spectral behaviour of the Laplacian matrix. The obtained spectrum can therefore be used for efficient retrieval, as the Laplacian spectrum has been proven to be reliable for this purpose in recent studies [11, 4]. Finally, in Section 4, experiments show the increase in representational power of the encoding over existing methods.

2. Hermitian spectral representation

One of the most natural and informative algebraic structures to associate with a graph is its Laplacian matrix. This matrix is defined as $L(G) = D(G) - A(G)$, where $D(G)$ is the diagonal matrix containing node degrees, and $A(G)$ is representing G's connectivity; the entry $A_{i,j}$ is 1 if nodes i and j are connected, 0 otherwise. As a result, for all rows in $L(G)$ the sum of the entries is 0. The spectrum of the Laplacian matrix can be used as a signature representation for the graph, and thus for the model that is represented by the graph. This signature representation can be used for efficient retrieval purposes (indexing), [4]. One of the main reasons for this is that many graph properties and invariants are implicitly or explicitly reflected by the Laplacian spectrum [8]. Moreover, cospectrality for non-isomorphic graphs tends to be rare [13] and similar Laplacian matrices have similar spectra due to the interlacing theorem for two graphs where one is a slightly modified version of the other [7].

In the case of a weighted graph, $L_w(G) = D_wG - A_wG$ can be obtained. In this case, $D_w(G)$ is a diagonal matrix containing for each node the sum of edge weights of its incident edges. Correspondingly, in the adjacency matrix the entry $A_{i,j}$ represents the weight associated with nodes i and j, which is 0 if there is no connecting edge between them. Therefore, all information with respect to the graph's connectivity is still present in $L_w(G)$, since every non-zero entry indicates the existence of an edge between the corresponding nodes.

In order to preserve the useful properties of a normal Laplacian spectrum, every edge weight $w_{a,b}$ should satisfy the following conditions:

$$\begin{array}{rcll} W_{a,b} & = & W_{b,a}, \text{ where } a,b \in V & (1) \\ W_{a,b} & \geq & 0, \text{ where } a,b \in V & (2) \\ W_{a,b} & \neq & 0, \text{ iff } a \text{ and } b \text{ are adjacent in } G & (3) \end{array}$$

Equation 1 ensures a symmetric matrix, whereas equation 3 ensures that the connectivity of the graph remains unchanged after weighting the edges.

Unfortunately, it is not possible to store more information in a Laplacian matrix than the graph's connectivity together with the edge weights. As a consequence, a spectral representation using this matrix will suffer in most cases from significant information loss, since other graph characteristics such as node labels, node locations (planar graphs, 3D graphs) or additional edge measurements are not captured by the encoding.

Therefore, following the ideas of [12], we use a Hermitian matrix to store graph characteristics. However, to really mimic the spectral behaviour of a Laplacian matrix, we added two additional constraints to the construction of the Hermitian matrices. First, we give a brief theoretical background on Hermitian matrices, and then we impose the constraints for mimicking Laplacian spectral properties.

A Hermitian matrix H (or self-adjoint matrix) is a square matrix with complex entries that is equal to its own conjugate transpose. In other words, $H_{i,j}$ is equal to the complex conjugate of $H_{j,i}$. Fortunately, every Hermitian matrix has a real valued spectrum. The corresponding eigenvectors however contain complex entries. By adding several additional constraints to the construction of H, we can mimic the spectral behaviour of a Laplacian matrix, i.e. we can construct a property matrix $H(G)$ for $G = (V, E)$ in such a way that we can use its spectrum for retrieval purposes equally well as the Laplacian spectrum.

To this end, the off-diagonal elements of H are chosen to be complex numbers written in polar form using Euler's formula:

$$H_{a,b} = -W_{a,b} e^{iy_{a,b}} \tag{4}$$

where each edge has the pair of observations $(W_{a,b}, y_{a,b})$. The second observation, represented as the phase of the complex matrix entry, must satisfy the following conditions:

$$y_{a,b} \quad = \quad -y_{b,a} \tag{5}$$
$$-\pi \quad < y_{a,b} < \quad \pi \tag{6}$$

The first condition (5) ensures that H is equal to its own conjugated transposed matrix. By obeying the second constraint (6), phase wrapping can be avoided.

The on-diagonal entries (that are required to be real) are chosen to be

$$H_{aa} = \sum_{b \neq a} W_{a,b} \tag{7}$$

In this way, the entries in each row of the matrix now sum up to zero. This on-diagonal entry is necessary, because all edge weights $W_{a,b}$ (magnitudes) are inserted as $-W_{a,b}$, see (4). By summing up the edge weights and inserting this sum as on-diagonal entry, the sum of the entries in each row is zero. We would like to stress that this is a necessary property to correctly mimic the spectral behaviour of Laplacian matrices, contrary to the Hermitian matrix that is used in [12] (where additional node measurements on the diagonal are allowed). Furthermore, edge weights (magnitudes of the complex entries) should be calculated in such a way that an edge between two nodes can never be weighted 0, for it would destroy the connectivity of the graph.

2.1 Retrieval based on spectra

It is the key function of any indexing algorithm to speed up the retrieval process by selecting a small set of candidate models that are either presented to the user, or passed on to a more refined matching unit in the retrieval pipeline. The representation used here during indexing is a spectral one, which is basically a d-dimensional vector of features where d is the number of nodes in the graph, or the size of the Hermitian matrix. Therefore, to evaluate similarity between two objects, we calculate the Euclidean distance between their feature sets, i.e. between their Hermitian spectra. When trademarks are of different size, the spectra are of different dimension. There are several ways to deal with this problem. It is possible to enlarge the spectrum of the smaller trademark by inserting zeros. This is semantically correct, since it means isolated nodes are added to the graph. Another possibility is to decompose the graph into several subgraphs, and match only subgraphs of the same size. For more details on how to handle graphs of different sizes, we refer to [11]. In the rest of this paper we will assume graphs are of the same size.

In order to index a large data set efficiently, the vectors can be accessed through a Balanced-Box-Decomposition Tree (BBD-Tree), as introduced in [1]. This data structure is proven to be optimal for $(1 + \epsilon)$-approximate nearest neighbour searching [1], where k approximate nearest neighbours in a d-dimensional space can be reported in $O(kd \log n)$ time.

3. Graph attributes

With the goal to describe a trademark, we construct the graph whose nodes represent the shapes of the trademark revealed after the segmentation phase. We connect each node with its six nearest neighbours based on the distance between the barycenters of the corresponding shapes. There are many possible attributes that can be used to enrich a graph structure with additional shape information. To name a few, the attributes can be the area, perimeter, curvature of the corresponding segment, whereas the edges can be weighted with the distance or the angle between the shapes. The scope of our work is to represent the layout of the trademark. To this end, we will use the information about the location and intersection of shapes with respect to each other. Moreover, the use of the Hermitian matrix for the graph encoding imposes the constraints (1)-(3), (5) and (6) which the graph attributes should satisfy.

3.1 Directional attributes

For the description of the position of one shape with respect to the other we chose the angular measure. Precisely, we compute the angle between the two lines formed by the end points of an edge and the barycenter of the trademark. See Figure 3 for an example. This attribute satisfies the conditions (5) and (6) and thus can be used as the phase of the complex off-diagonal entries of the Hermitian matrix.

3.2 Topological attributes

Egenhofer and Franzosa [5] pointed out that there are 8 basic topological relations: disjoint, contains, inside, meet, equal, covers, covered-by and overlap. These relations, or intersection types, can be partially captured with one intersection measure on two components, which we define as

$$W_{ab} = \frac{Area_{ab}}{Area_a + Area_b}$$

[1] An object is a $(1 + \epsilon)$-approximate k-nearest neighbour of the query if its distance to the query is within a factor of $(1 + \epsilon)$ to the distance between the query and its true k-nearest neighbour.

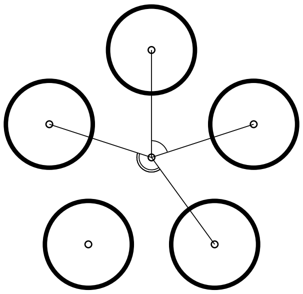


Figure 3. Computation of directional attributes: angle between lines formed by connecting the two end points of each edge to the trademark's barycenter.

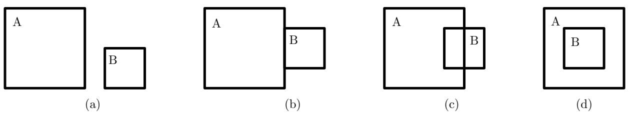


Figure 4. Different intersection types for the shapes $Area_A = 4$, $Area_B = 1$. (a) separate shapes $W_{AB} = 1$, (b) touching shapes $W_{AB} \approx 1$, (c) intersecting shapes $W_{AB} = 0.91$, (d) shape a includes shape b $W_{AB} = 0.8$.

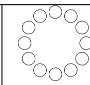
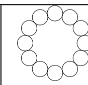
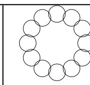
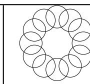
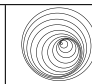
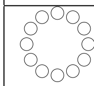
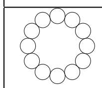
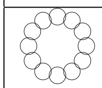
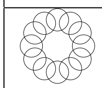
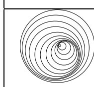
where $Area_{ab}$ is the area of the union of the components a and b. The area of a component is measured by the number of pixels occupied including boundary pixels. For two separated segments the intersection measure is equal to one and decreases as the intersection area increases. Figure 4 illustrates different types of the intersections. The intersection measure satisfies the conditions (1)-(3) and thus can be used as the magnitude of the complex elements of the Hermitian matrix.

4. Experiments

To evaluate the effectiveness of the proposed approach, we focus on comparing several typical examples of shape configuration. Therefore, in this section we will assume that segmentation reveals the individual shapes, and calculates the angular and topological values. Furthermore, we will assume that the graphs that are compared are of the same size, i.e. they have the same number of vertices. For details on an appropriate segmentation technique, and on how to work with graphs of different sizes we refer to a recent study [11].

In this Section, we will evaluate how distances between pairs of trademarks change when topological and directional changes in the configuration occur. At this point, we would like to point out that every distance will be 0, should each image be represented by the spectrum of its normal Laplacian matrix. When a weighted Laplacian matrix is chosen as associated structure, angular or directional changes in the configuration are not revealed during similarity evaluation.

Table 1. Distance matrix for different topological configurations of 12 circles.

dist					
	0	0.023	0.107	1.067	5.441
	0.023	0	0.084	1.044	5.420
	0.107	0.084	0	0.960	5.343
	1.067	1.044	0.960	0	4.474
	5.441	5.420	5.343	4.474	0

In the first experiment, all pairwise distances between 5 configurations of 12 circles are calculated. The angles between the circles are the same in all images, the overlap varies from disjoint to touching, overlapping, more overlapping and inclusion. See Table 1 for the results of this experiment together with the images that are used for calculation. The results clearly show how distances increase when overlap increases. The experiment is repeated with configurations of four squares in Table 2. Again, the angles between the squares remain constant, while the overlap increases from no overlap to inclusion. For these examples, distances grow proportionally with increasing overlap as well. Furthermore, this experiment shows that calculation of the topological attributes is dependent on the shape of the components. For instance, a larger distance is found for configurations of squares than of circles between a disjoint configuration and a touching configuration (first row, second column of both Tables 1 and 2).

The third experiment, of which the results are given in Table 3, shows the benefit of the directional information in the encoding. All these distances would have been 0 using normal or even weighted Laplacian matrices as a representation. The distances listed in Table 3 coincide with the perceived similarity between the images. For example, the first and the third images both appear to have a smaller distance to each other than to all other images, which is a desired result in this case.

The images used for the final experiment have variations in both topological and directional configuration. As the results show in Table 4, even with these combined alterations, distances reflect the similarity in layout. Take for instance the pair of the first and fourth images, that have a closer distance to each other than to all other images. Furthermore, the influence of the enclosing frame in the fifth image is clearly present, since it has a large distance to all other models. Finally, the two images containing only

Table 2. Distance matrix for different topological configurations of 4 squares.

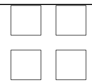
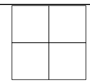
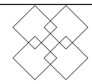
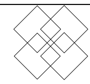
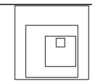
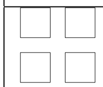
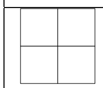
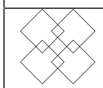
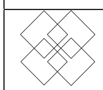
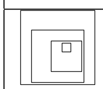
dist					
	0	0.036	0.167	0.489	1.046
	0.036	0	0.161	0.453	1.014
	0.167	0.161	0	0.333	0.899
	0.489	0.453	0.333	0	0.643
	1.046	1.014	0.899	0.643	0

Table 3. Distance matrix for different angular configurations of 5 circles.

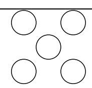
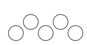
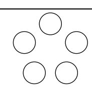
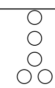
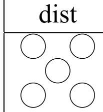
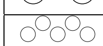
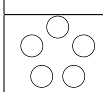
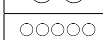
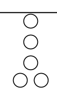
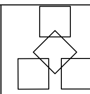
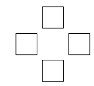
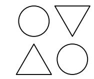
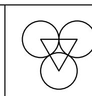
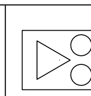
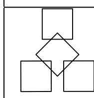
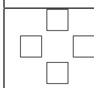
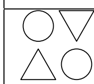
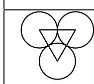
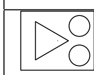
dist					
	0	1.119	0.531	1.933	1.24
	1.119	0	0.917	0.875	0.237
	0.531	0.917	0	1.782	1.108
	1.933	0.875	1.782	0	0.716
	1.24	0.237	1.108	0.716	0

Table 4. Distance matrix for different configurations 4 of shapes with mixed properties.

dist					
	0	0.446	0.922	0.173	1.446
	0.446	0	0.582	0.513	1.208
	0.922	0.582	0	0.902	0.695
	0.173	0.513	0.902	0	1.356
	1.446	1.208	0.695	1.356	0

disjoint components (second and third image) are close to each other, but still have a nonzero distance because of differences in directional attributes.

5. Conclusion

In this paper we have presented a new approach for encoding layout between image components that, together with the shapes of the components, is important for evaluating similarity between images. Both directional and topological relations between image components that are near each other, are encoded in a rich graph structure. By associating a Hermitian matrix to the graph, and by obeying several constraints on the computation of edge weights, we are able to capture more edge information (together with the connectivity) in a spectral representation that mimics the behaviour of Laplacian spectra. Therefore, similarity evaluation is efficient and accurate, and the proposed approach can be successfully applied as an indexing mechanism.

The next step will be to evaluate the new approach within the context of a real retrieval environment. Although it was shown before that spectral representations are well suited for this kind of retrieval purposes, and we have shown in this paper that the new Hermitian spectral representation is more discriminating and provides distance values that reflect layout similarity better, it is important to investigate retrieval performance on a real data set of trademark images. To do so, we will make use of a large collection of real trademark images that has been classified by trademark experts who evaluate trademark similarity on a daily basis. We will compare our results to other methods using popular and representative performance measures such as Average Dynamic Precision, Mean Cumulative Gain Vectors and Mean Discounted Cumulative Gain Vectors.

Furthermore, it is one of our interests in the near future to explore part-based similarity between graphs using a spectral approach. Since the eigen decomposition of a Hermitian matrix reveals the eigenvectors as well as the spectrum, we automatically obtain the eigenvector associated with the second smallest eigenvalue (the so-called Fiedler vector and Fiedler value respectively) [8]. The Fiedler vector can be used for partitioning the graph in sensible parts, avoiding the computationally expensive inspection of all possible subgraphs of all different sizes. These subgraphs can be represented again by their Hermitian spectra. A voting schema will be necessary to combine search results for complete and partial graphs.

Acknowledgements

This research was supported by FP6 IST projects 511572-2 PROFI and 506766 AIM@SHAPE.

References

[1] S. Ayra, D.M. Mount, N.S. Netanyahu, R.Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM, 45(6):891–923, 1998.

[2] S.K. Chang, E. Jungert, and Y. Li. Representation and retrieval of symbolic pictures using generalized 2D strings. In Conference on Visual Communications and Image Processing, volume 3, pages 1360–1372, November 1989.

[3] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2-d strings. Pattern Analysis and Machine Intelligence, 9(3):413–428, 1987.

[4] M.F. Demirci, R.H. van Leuken, and R.C. Veltkamp. Indexing through laplacian spectra. Computer Vision and Image Understanding, In press, 2007.

[5] M.J. Egenhofer and R.D. Franzosa. Point Set Topological Relations. Geographical Information Systems, 5(2):161–174, 1991.

[6] S.Y. Lee and F.J. Hsu. 2D C-string: a new spatial knowledge representation for image database systems. Pattern Recognition, 23(10):1077–1087, 1990.

[7] R. Merris. Laplacian matrices of graphs: a survey. Linear Algebra and its Applications, 197(1):143–176, 1994.

[8] B. Mohar. The laplacian spectrum of graphs. Graph Theory, Combinatorics and Applications, 2:871–898, 1991.

[9] E.G.M. Petrakis and S.C. Orphanoudakis. A Methology for the Representation, Indexing, and Retrieval of Images by Content. Image and Vision Computing, 8(11):504–512, October 1993.

[10] J. Schietse, J.P. Eakins, and R.C. Veltkamp. Practice and challenges in trademark image retrieval. In Conference on Image and video retrieval, pages 518–524, 2007.

[11] R.H. van Leuken, M.F. Demirci, V.J. Hodge, J. Austin, and R.C. Veltkamp. Layout indexing of trademark images. In Conference on Image and video retrieval, pages 525–532, 2007.

[12] R.C. Wilson, E.R Hancock, and B. Luo. Pattern vectors from algebraic graph theory. Pattern Analysis and Machine Intelligence, 27:1112–1124, 2005.

[13] P. Zhu and R.C. Wilson. A study of graph spectra for comparing graphs. In British Machine Vision Conference, 2005.