Local Binary Patterns for Graph Characterization

Muhammad Jawad
Institute of Management Sciences,
Peshawar, Pakistan
Email: mjawad798@gmail.com

Furqan Aziz
Institute of Management Sciences,
Peshawar, Pakistan
Email: furqan.aziz@imsciences.edu.pk

Edwin Hancock
Department of Computer Science,
University of York, YO10 5GH, UK
Email: edwin.hancock@york.ac.uk

***Abstract*—In this paper we propose a novel approach for defining Local Binary Patterns (LBP) to directly encode graph structure. LBP is a simple and widely used technique for texture analysis in static 2D images, and there is no work in the literature describing its generalisation to graphs. The proposed method (GraphLBP) is efficient and yet effective as a noise-tolerant graph-based representation. We compute the new feature representation for graphs by combining LBP with Galois Fields, using irreducible polynomials. The proposed method is scalable as it preserves the local and global properties of the graph. Experimental results show that GraphLBP can both increase the recognition accuracy and is both simpler and more computationally efficient when compared with state of the art techniques.**

***Index Terms*—Graph Characterization, Local Binary Patterns, Galois Fields**

I. Introduction

There has recently been an increasing interest in how to analyze and compare patterns represented using graphs. This is due to the richer representational power offered by structures such as tree, graphs and hypergraphs. Moreover the possibility of using such data representations, places considerable demands on the available methodology from machine learning and pattern recognition, which usually operate manily on vectorial data. A graph represents a pattern where nodes represent features and the edges represent their relationships. Over the past two decades graph-based methods have been widely used to model and solve problems in different domains. For instance a two-dimensional image can be represented by a planar graph whose nodes represent pixles or pixel features and the edges represent spatial relationship between those features. A mesh, on the other hand, provides a reliable representation of a shape represented in a three-dimensional (3D) space.

However, one of the limitations of graphs is that they cannot be directly used for analysis tasks. For example, a mesh constructed over a 3D shape might be very useful for visualisation tasks but it may not be useful for shape retrieval task. This is due to the lack of natural ordering in the vertices (or edges) of the graph and so the traditional statistical pattern recognition techniques cannot be directly applied to graphs. Therefore, compared to feature vectors, graphs based methods usually have high complexities. For example, while comparing two vectors for equality can be done in linear time with respect to the length of vectors, comparing two graphs for exact similarity is not known to be in P class till day. Another difficulty with graph-based representation is the high sensitivity of graph to noise. Ideally the graph-based methods must be tolerant and should accommodate the noise by relaxing the graph matching constraints. For these reasons, exact algorithms may not be practical.

To address the difficulties with graphs, an number of approximate graph-based methods have been proposed and successfully applied in different domains. These methods can be broadly divided into two categories, i.e., inexact methods and decomposition methods. The first approach, Inexact methods, include the use of graph-edit distance to embed the graph in high-dimensional feature space. These methods first define a set of graphs (called graphlets) that act as the bases set for graph embedding [1]. A graph is embedded to a high dimensional feature space by computing its graph edit distance from each of the bases graphs. In the decomposition methods the graph are first decomposed into substructures and then the frequencies of these substrucutres are used to embed the graphs into high dimensional feature space. A number of approaches have been used to decompose a graph into substructures. The simplest one uses the topological properties of a graph i.e. vertex number, edge number, diameter etc. These methods are computationally very efficient but are usually not very expressive. For example in [2] Dutta et al. have shown that out of 79 possible graphs of size 7, 44 collisions were detected when they were pairwise compared for degree distribution. In another approach, graphs are decomposed into smaller subgraph based on structural properties such as walks [3], paths [4], cycles [5] and trees [6] etc. These methods provide a more expressive representation of graph but are computationally expensive. To improve the performance of decomposition methods, graph kernels have been introduced. A graph kernel is a function that computes the similarity between graphs by assuming an implicit embedding of the graphs in a high-dimensional feature space instead of decomposing graphs into substructures. Finally spectral methods have also been successfully used that uses the spectrum of Laplacian or Adjacency matrix representation of a graph. These methods have been proved successful for correspondence matching and clustering of both 2D images [7] and 3D shapes [8]. However those usually require at least cubic time in size of the graph.

In this paper we propose a novel decomposition framework to embed the graphs in a high dimensional feature space. The advantage of our approach is that it is based on the degree distribution of a graph and is computationally very efficient. We also empirically show that the proposed method achieves higher accuracy when compared to state-of-the-art decomposition techniques that are based on the structural properties of a graph. Our idea is inspired by the Local Binary Pattern (LBP) that was originally proposed by [9] for texture analysis of 2D images. Due to its computational simplicity and discriminating power, it attracted the pattern recognition and image processing researchers and also found its application in other areas like remote sensing [10], visual inspection [11], face recognition [12] & motion analysis [13]. Recently, Werghi et al.[14] proposed a novel framework based on the idea of LBP for texture analysis of 3D shapes that are represented by meshes. The advantage of LBP-based methods for features extraction over traditional approaches is that of its simplicity and effectiveness. Motivated by this, in this paper we propose a novel framework, referred to as *GraphLBP*, that extends the idea of LBP to graphs. Our aim is to capture the dominant features of the vertices with its neighbours and encode the local structure around each vertex. To obtain a small set of the most discriminative LBP-based features for better performance and dimensionality reduction, LBP-based representations are associated with Galois Field Algebra which are useful in translating the local features into a vector of fixed length.

II. Preliminaries

In this section we will give some basic definitions of important terminologies which are used throughout the paper.

A. Graph

A *graph* $G = (V, E)$ consists of a finite nonempty set of *vertices* V and a finite set of *edges* E. Two vertices v_i and v_j are neighbours or $adjacent$ if they are the end vertices of the same edge $e_k = (v_i, v_j)$. Two edges e_i and e_j are adjacent if they have an end vertex in common, say v_k, i.e. $e_i = (v_k, v_l)$ and $e_j = (v_k, v_m)$. If all vertices of G are pairwise neighbours, then G is complete. An edge is called $incident$ on its end vertices. The degree (or valency) $deg(V)$ of a vertex V is the number of edges incident on it.

B. Local Binary Patterns

Local Binary Patterns (LBP) are a non-parametric method, that summarises local image structures efficiently by comparing each feature of the object with its neighbouring features. The Local Binary Pattern (LBP) was introduced by Ojala et al. [15] [16] for describing 2D textures in still images. The most important properties of LBP for images are its tolerance regarding monotonic illumination changes and its computational simplicity. In the original definition, the LBP operator [15] assigns labels to image pixels by first comparing the 8 neighbours with the centre value (i.e., the neighbor pixel value is considered as 1 if its value is greater or equal to the central pixel value, and 0 otherwise), then considering the sequence of 1/0 in the pixel neighbourhood as a binary number. This is shown in Figure 1, where the upper left pixel in the neighbourhood is regarded as the most significant bit in the final code. This eight bit number encodes the mutual relationship between the gray levels of the central pixel and its neighbouring pixels. The histogram of the numbers obtained in such a way can then be used as a texture descriptor. This operator distinguished by its simplicity and its invariance to monotonic gray-level transformations.

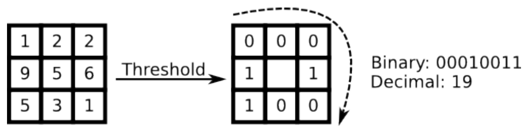


Fig. 1. Computation of the basic LBP code from 3 x 3 neighbourhood of a central pixel. The central pixel is compared with each neighbour, starting from upper-left corner and produce 1 if its value is greater or equal, 0 otherwise. The result is an 8-bit binary code

LBP can be extended to operate on circular neighbourhoods of different radii, allowing sub-pixel alterations [16]. These initial formulations subsequently led to the definition of alternative neighbourhood variants. For instance, Liao et al. [17] proposed oriented neighbourhood LBP which accounts for anisotropic information. Similarly the multi-block LBP(MB-LBP) that compares the averages of the gray level intensity of neighbouring pixels rather than the value of individual pixels, in order to capture macrostructural features in the image [18]. A more complete list and discussion on the many LBP variants can be found in [19].

C. Galois Field Algebra

A Galois Field is a finite field, i.e., a field in which there exists finitely many elements. For Galois Fields, the order of the field (i.e., the number of elements in the field) is always a prime or a power of a prime. For any prime integer p and any integer m greater than or equal to 1, there is a unique field with p^m elements denoted as $GF(p^m)$. These finite fields are extensively used in cryptographic algorithms like Advanced Encryption Standard(AES), elliptical Curve Cryptography(ECC) as well as in coding theory like Reed Solomon codes. It is particularly useful in translating computer data as they are represented in binary forms. Representing data as a vector in a Galois field allows mathematical operations to scramble data easily and effectively. In this paper our goal is to use Galois field on the binary patterns obtained from the LBP when applied on a graph. Since our data is represented in the form of binary numbers, so we will assume our binary patterns are elements of $GF(2^m)$. As with any other field, the basic operations are defined in Galois field. Two most commonly used operations are *multiplication* and *addition*.

Addition in Galois Fields: In $GF(2^m)$, addition is especially easy, since addition and subtraction is the same, and furthermore this operation can be done in hardware using basic XOR logic gate, since there is no concept of carry generation and carry propagation.

Multiplication in Galois Fields: In $GF(2^m)$, multiplication is performed using polynomial multiplication followed by modular reduction using polynomial. In our case we are doing modular reduction via irreducible polynomial. A polynomial is said to be irreducible if it cannot be factored into nontrivial polynomials over the same field. For example in the field of rational polynomials $Q[x]$ (i.e., polynomials $f(x)$ with rational coefficients), $f(x)$ is said to be irreducible if there do not exist two non-constant polynomials $g(x)$ and $h(x)$ in x with rational coefficients such that $f(x) = g(x)h(x)$. A list of irreducible polynomials of degree 2 to 5 is given in Table I.

TABLE I
IRREDUCIBLE POLYNOMIALS OF DEGREES 2 THROUGH 5

Degree	irreducible polynomials
2	$1 + x + x^2$
3	$1 + x + x^3$, $1 + x^2 + x^3$
4	$1 + x + x^4$,$1 + x + x^2 + x^3 + x^4$, $1 + x^3 + x^4$
5	$1 + x^2 + x^5$, $1 + x + x^2 + x^3 + x^5$, $1 + x^3 + x^5$, $1 + x + x^3 + x^4 + x^5$, $1 + x^2 + x^3 + x^4 + x^5$, $1 + x + x^2 + x^4 + x^5$

III. GRAPHLBP

In this section we describe how LBP can be defined for a graph. We discuss the challenges that are involved in defining LBP on a graph and propose methods to overcome those problems. We refer the proposed framework as GraphLBP. We begin by defining LBP In its original form, the LBP operator assigns labels to image pixels by comparing the intensity value of a pixel with its 8 neighbours and is given by

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c)2^p,$$

where g_c is the gray value of the central pixel, g_p is the gray value of its neighbours and P is the total number of involved labels. The value of the function $s(x)$ is 1 if $x \geq 0$ and 0 otherwise.

In our approach, we define LBP for every vertex of a graph. For a labelled graph, where every vertex of a graph is assigned a unique label, comparison can be done directly (if there exists a partial ordering between lables). For unlabelled graphs, we use the degree of a vertex to construct LBP, i.e, the degree of a vertex is compared with the degree of its neighbour vertices. However, applying the LBP on graph-based representation is not a straight forward method because the graph-based representation has few limitations. First, there is no ordering information available in the vertices of a non-planner graph. This will result in different LBP for different ordering of neighbouring vertices. Secondly, the number of neighbours of a vertex are not fixed, resulting in LBP with varying lengths. Finally, graphs are sensitive to noise due to which there are additional/missing edges/vertices. For these reasons, applying LBP operator directly may not be practical.

To overcome these problems, in this section we propose two algorithms, that together can be used to define GraphLBP. We begin by defining a LBP for a vertex of a graph. As mentioned earlier, we use the degree of a vertex to define its local binary pattern. To construct LBP for a vertex v, we take all the neighbours of v and sort them in descending order according to their degrees. The pattern value for each vertex v is computed by comparing its degree with the degrees of its neighbour vertices and produce 1 if the $deg(v)$ is greater or equal to the degree of its neighbour, otherwise 0. Consider, for example, the graph of Figure 2.

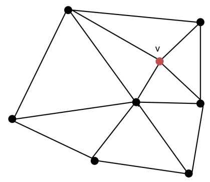


Fig. 2. A simple graph with 8 vertices

To construct a LBP for the vertex v in the graph of Figure 2, we take all its four neighbours and sort them into degree sequence order. The resulting sorted sequence is $6, 4, 4, 3$. Since $deg(v)$ is 4, the LBP for the vertex v is 0111. This procedure is outlined in Algorithm 1.

Algorithm 1 Local Binary Patterns of Graph Vertices

Input : Graph $G = (V, E)$
Output: G_{LBP} ▷ Local Binary Pattern of each vertex of graph
procedure GRAPHLBP(G)
  for $i \leftarrow 1, all\_vertices$ **do**
    $V_{nieghbors} \leftarrow Get\_Neighbors(V_i)$ ▷ Get Neighbors of each Vertex
    $N_{sorted} \leftarrow Sort\_Neighbors(V_{neighbors})$ ▷ Sort the neighbor vertices w.r.t highest degree
    for $j \leftarrow 1, all\_Neighbors$ **do** ▷ Comparing the vertex with its neighbors
      if $degs(V_i) >= N_{sorted(j)}$ **then**
        $G_{LBP(i,j)} \leftarrow 1$
      else
        $G_{LBP(i,j)} \leftarrow 0$
      end if
    end for
  end for
end procedure

Note that the resulting binary pattern produced for a vertex v by Algorithm 1 is not of fixed length and encodes only information local to the vertex v. To obtain a fixed length encoding and to define a stronger representation for a graph, we combine the local binary pattern of a vertex with its neighbours and make use of the Galois Field. This is done by adding the LBP of a vertex with its neighbours using field addition. To obtain a fixed length encoding, the resulting value is reduced using an irreducible polynomial of a fixed degree. This will produce a binary hash value for each vertex of a graph. To understand this, consider the vertex v of graph of Figure 2. The LBP produced by Algorithm 1 for v is 0111, while LBP produced for its neighbours are $111111, 0011, 0011, 000$. Adding all these values via $galoisfieldaddition$ will produce 111111. Reducing the binary values using, for example, the irreducible polynomial $1 + x + x^3$ of degree 3 will produce a hash value

110. This value will be treated as the LBP of the vertex v. Note that this approach has two advantages. Firstly, it produces fixed length codes for each vertex. Secondly, it produces a more richer encoding by incorporating the information of the neighbours. The local binary patterns of the vertices are finally grouped using histogram binning to produce a global signature for graph characterization. Algorithm 2 outlines the steps performed in computing GraphLBP.

Algorithm 2 Features Extraction from Graph

Input : Graph G_{LBP}, n
  ▷ The LBP computed in Algorithm 1 and bin size
Output: $Graph_{Vector}$ ▷ Feature Vector of the Graph
procedure GETVECTOR(G_{LBP})
  for $i \leftarrow 1, all\_vertices$ **do**
    for $j \leftarrow 1, all\_Neighbors$ **do**
      $V_{add} \leftarrow gfadd(G_{LBP}(Neighbor_j), V_{add})$
    end for
    $G_{vec}(i) \leftarrow gfdeconv(V_{add}, P_{irreducible})$
  end for
  $Graph_{vector} \leftarrow hist(G_{vec}, 10)$
end procedure

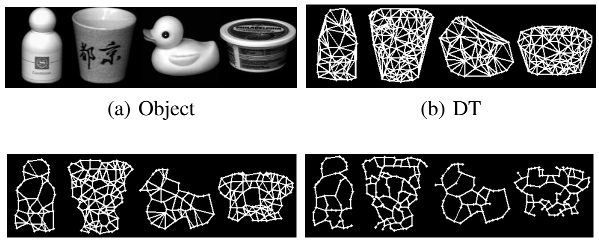
Note that the Algorithm 2 requires two external parameters. In our experimental evaluation, we have chosen the irreducible polynomial $1 + x + x^3$ of degree 3, while the number of bins as 10.

Time Analysis: The worst case running time of the GraphLBP (Algorithm 2) is $O(|V|^2)$. This is due to the fact that, in the worst case(assuming complete graph), both the outer loop and the inner loop in the algorithm will be executed and will take $O(|V|^2)$ of running time. For a graph, represented in the form of adjacency list, the aggregate running time of the algorithm is $O(|E|)$. Note that the running time of most state of the art algorithms including random walk, Ihara coefficients, and shape DNA is $O(|V|^3)$.

IV. EXPERIMENTAL EVALUATION

In this section we perform experimental evaluation of the proposed method and compare it with state of the art methods. For this purpose, we selected the graphs that are extracted from different views of an object taken with various transformation and illumination conditions. The objective is to assess whether GraphLBP can be used to embed the graphs in a vector space to characterize their structure. The images are selected from **COIL (Columbia Object Image Library) [20]**. This dataset consists of 20 different objects each with 72 views. These views are obtained from equally spaced directions over 360^o. In our experiments we have selected 4 different objects with all their 72 views. Figure 3a shows some examples of photographs taken from COIL.

To construct graphs over these images, we have applied **Harris corner detector** [21]. Harris corner detector is used to extract a list of candidate feature points. We treat these feature points as vertices and construct a Delaunay triangulation over those feature points. A **Delaunay triangulations (DT) [22]** for a set P of points in a Euclidean space is a triangulation, DT(P), such that no point in P is inside the circumcircle of any triangle in DT(P). Figure 3 shows an example of an object and its corresponding Delaunay Triangulation.



(a) Object (b) DT

(c) GG (d) RNG

Fig. 3. COIL Objects and their extracted Graphs

Once the graphs are extracted from the object images, we apply GraphLBP to the extracted graphs to embed them in a high-dimensional feature space. To evaluate the performance of the proposed method, we compare it with following state-of-the-art methods.

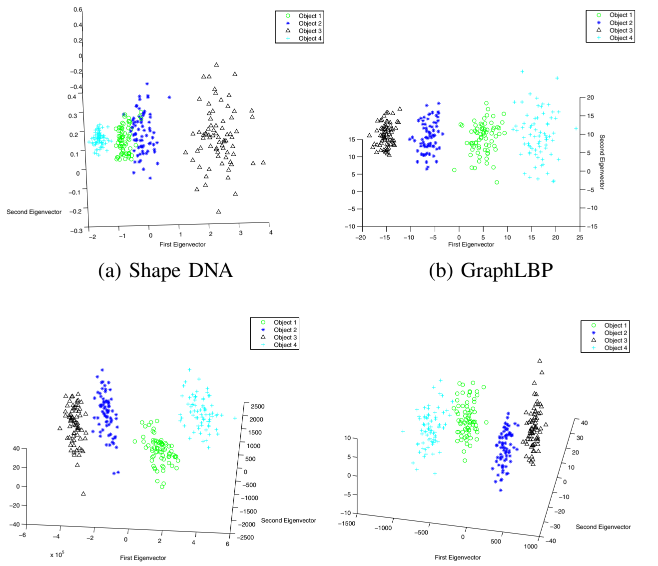
Random walk kernel [3]: Random walk kernel is state-of-the-art graph kernel used to compare graphs. It measures the similarity between two graphs by counting the frequencies of matching random walks in the two input graphs. It avoids the decomposition of the input graphs in to walks by using the product graph formalism. This increases the efficiency of the kernel.

shape DNA [23]: This method defines shape by a vector composed of first few smallest eigenvalues of the Laplacian matrix representation of a graph. This method was originally proposed by Reuter et al. [23] for 3D shape classification. For our experiments, we have chosen first ten positive eigenvalues of the Laplacian matrix. Note that the smallest eigenvalue of the Laplacian matrix is always zero and so we have ignored it in our representation.

Ihara coefficients [5]: This method uses a feature-vector that records prime cycle frequencies in a graph. These cycle frequencies are computed using first few coefficients of the reciprocal of the Ihara zeta function of the graph, commonly referred to as Ihara coefficients. For comparison purpose in our paper, we use the feature vector constructed from the coefficients c_3, c_4 and $c_{ln|2E|}$, as proposed by Peng in[5]. Note that Ihara coefficients are considered a powerful tool to capture the cyclic structure of graphs [5], [24].

Next we apply the these methods to the Delaunay triangulations extracted from all the three datasets. To compare the visualisation results, we apply principal component analysis (PCA) to the resulting feature vectors. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate(called the first principal component), the second greatest variance on the second coordinate, and so on. Figure 4 compares the visualization results on the first three principal components of Delaunay triangulations extracted from the

COIL dataset.



(a) Shape DNA (b) GraphLBP

(c) Random Walk Kernel (d) Ihara Coefficients

Fig. 4. PCA embedding of feature vectors computed from Delaunay Triangulations.

To quantitatively compare the performance of the proposed method with alternative methods, we cluster the graphs using k-means clustering [25]. k-means clustering is a method, which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. We compute Rand index [26] of these clusters, which is a measure of the similarity between two data clusters. Table II compares the Rand indices of all the methods.

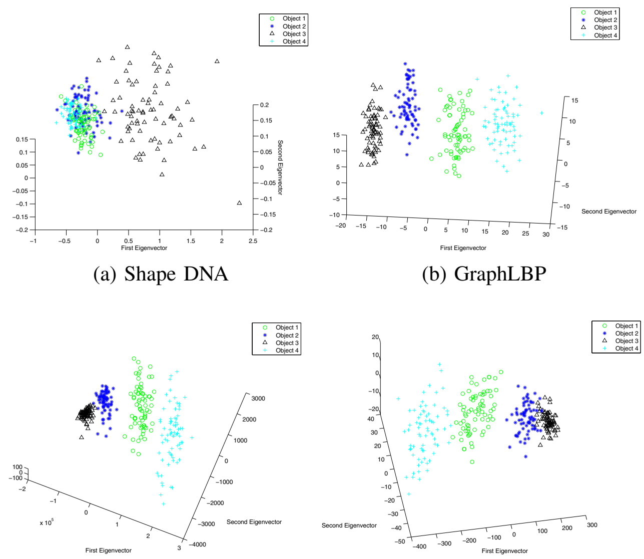
TABLE II
ACCURACIES OF DIFFERENT METHODS ON DELAUNAY TRIANGULATIONS

Method	**DT**
GraphLBP	99.65%
Shape DNA	97.36%
Ihara	97.96%
Random Walk Kernel	97.99%
Selected Ihara	98.97%

The above results show that GraphLBP can give better performance as compared to some state of the art methods.

To take this study one step further, we now apply GraphLBP to Gabriel graphs and relative neighbourhood graphs extracted from the same dataset. A **Gabriel Graph [27]** for a set of n points is a subset of Delaunay triangulation, which connects two data points v_i and v_j for which there is no other point v_k inside the open ball whose diameter is the edge (v_i, v_j). The **relative neighbourhood graph [28]** is also a subset of Delaunay Triangulation. In this case a lune is constructed on each Delaunay edge. The circles enclosing the lune have their centres at the end-points of the Delaunay edge; each circle has a radius equal to the length of the edge. If the lune contains another node then its defining edge is pruned from the relative neighbourhood graph. Figure 3c and 3d show an example of a Gabriel Graph and a relative neighbourhood graph respectively for corresponding images shown in Figure 3a. Note that, since both the GG and RNG are subset of DT, the experiments on those datasets allow us to investigate the performance of the proposed method under controlled structural modification.

As with DT, we apply GraphLBP and alternate methods to the graphs extracted from the same objects and embed the resulting feature vectors in a three dimensional vector space using PCA. Figure 5 shows the resulting embeddings of the feature vectors extracted from Gabriel graphs, while Figure 6 shows the resulting embeddings of the feature vectors extracted from Relative neighbourhood graphs. For comparison purpose, we have shown the visualisation results for the proposed method and the alternate methods.



(a) Shape DNA (b) GraphLBP

(c) Random Walk Kernel (d) Ihara Coefficients

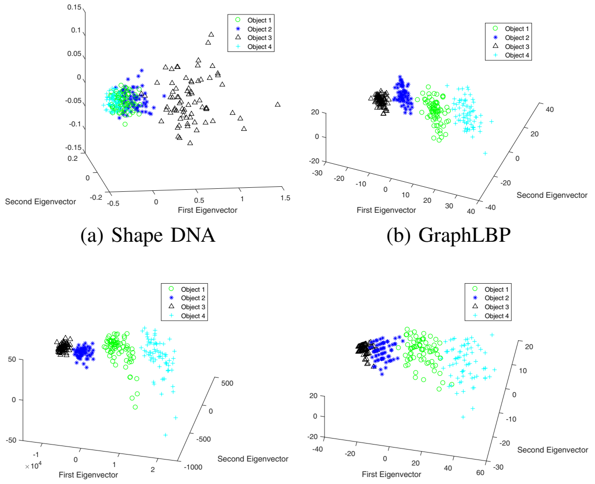
Fig. 5. PCA embedding of feature vectors computed from Gabriel graphs.

The embedding results of Figure 5 suggest that, under controlled structural modification, GraphLBP can still provide a better separation as compared to other state of the art methods. To quantitatively evaluate the performance of the proposed method, we compute the Rand index of the resulting clusters. Table III reports the resulting Rand indices for the proposed method and alternative methods.

TABLE III
ACCURACIES OF DIFFERENT METHODS ON GABRIEL GRAPHS

Method	**GG**	**RNG**
GraphLBP	**99.65%**	**98.30%**
Shape DNA	71.66%	75.06%
Ihara	82.31%	64.97%
Random Walk Kernel	93.27%	95.66%
Selected Ihara	93.85%	95.34%

It is clear from the table III that under controlled structural modifications, the proposed method still gives superior performance when compared to alternate methods.



(a) Shape DNA (b) GraphLBP

(c) Random Walk Kernel (d) Ihara Coefficients

Fig. 6. PCA embedding of feature vectors computed from relative neighbourhood graphs.

V. Conclusion

In this paper we have described Graph-LBP. This is a novel framework for characterizing graphs extracted from both real world and synthetic data. The proposed method is scaleable and maintains the simplicity and elegance characteristics of original pixel LBP, which can be used to construct the feature vectors from image textures. We provided a route for extracting structural properties from graphs via LBP, and have mapped them to a vector space using Galois field algebra. Our future research directions will focus on a) expanding our method to other datasets i.e. ALOI, ETHZ etc. b) encompass weighted graphs, directed graphs and hyper graphs - so that it can be extended to develop more general object representations, and c) expand of idea of Graph-LBP on 3D mesh models.

References

[1] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recognition Letters*, vol. 18, no. 8, pp. 689–694, 1997.

[2] A. Dutta and H. Sahbi, "High order stochastic graphlet embedding for graph-based pattern recognition," *arXiv:1702.00156*, 2017.

[3] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 129–143.

[4] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[5] P. Ren, R. C. Wilson, and E. R. Hancock, "Graph characterization via ihara coefficients," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 233–245, 2011.

[6] J. Qiangrong, L. Hualan, and G. Yuan, "Cycle kernel based on spanning tree," in *Electrical and Control Engineering (ICECE), 2010 International Conference on*. IEEE, 2010, pp. 656–659.

[7] E. Hancock, "Spectral approaches to learning in the graph domain," in *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 47–47.

[8] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comp. Grarph Forum*, pp. 1383–1392, 2010.

[9] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, 1994, pp. 582–585 vol.1.

[10] C. Song, F. Yang, and P. Li, "Rotation invariant texture measured by local binary pattern for remote sensing image classification," in *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, vol. 3. IEEE, 2010, pp. 3–6.

[11] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using lbp variance (lbpv) with global matching," *Pattern recognition*, vol. 43, no. 3, pp. 706–719, 2010.

[12] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 765–781, 2011.

[13] L. Cai, C. Ge, Y.-M. Zhao, and X. Yang, "Fast tracking of object contour based on color and texture," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1421–1438, 2009.

[14] N. Werghi, S. Berretti, and A. Del Bimbo, "The mesh-lbp: a framework for extracting local binary patterns from discrete manifolds," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 220–235, 2015.

[15] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[17] S. Liao and A. C. Chung, "Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude," in *Asian conference on computer vision*. Springer, 2007, pp. 672–679.

[18] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block lbp representation," in *International Conference on Biometrics*. Springer, 2007, pp. 11–18.

[19] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer vision using local binary patterns*. Springer Science & Business Media, 2011, vol. 40.

[20] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," Technical report CUCS-005-96, Tech. Rep., 1996.

[21] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[22] B. Delaunay, "Sur la sphére vide," *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, pp. 793–800, 1934.

[23] M. Reuter, F.-E. Wolter, and N. Peinecke, "Laplacebeltrami spectra as shape-dna of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342 – 366, 2006.

[24] F. Aziz, R. C. Wilson, and E. R. Hancock, "Backtrackless walks on a graph," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 6, pp. 977–989, 2013.

[25] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press, 1967, pp. 281–297.

[26] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[27] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, pp. 205–222, 1969.

[28] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261 – 268, 1980.