



Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviuEfficient many-to-many feature matching under the l_1 normM. Fatih Demirci^{a,*}, Yusuf Osmanlioglu^a, Ali Shokoufandeh^b, Sven Dickinson^c^a TOBB University of Economics and Technology, Söğütözü, Ankara 06560, Turkey^b Drexel University, Philadelphia, PA 19104, USA^c University of Toronto, Toronto, Ontario, Canada M5S 3G4

ARTICLE INFO

Article history:

Received 1 December 2009

Accepted 1 December 2010

Available online xxx

Keywords:

Distortion-free metric embedding

Earth Mover's Distance

Many-to-many matching

Object recognition

ABSTRACT

Matching configurations of image features, represented as attributed graphs, to configurations of model features is an important component in many object recognition algorithms. Noisy segmentation of images and imprecise feature detection may lead to graphs that represent visually similar configurations that do not admit an injective matching. In previous work, we presented a framework which computed an explicit many-to-many vertex correspondence between attributed graphs of features configurations. The framework utilized a low distortion embedding function to map the nodes of the graphs into point sets in a vector space. The Earth Movers Distance (EMD) algorithm was then used to match the resulting points, with the computed flows specifying the many-to-many vertex correspondences between the input graphs. In this paper, we will present a distortion-free embedding, which represents input graphs as metric trees and then embeds them isometrically in the geometric space under the l_1 norm. This not only improves the representational power of graphs in the geometric space, it also reduces the complexity of the previous work using recent developments in computing EMD under l_1 . Empirical evaluation of the algorithm on a set of recognition trials, including a comparison with previous approaches, demonstrates the effectiveness and robustness of the proposed framework.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Feature matching is a fundamental problem in computer vision, and plays a critical role in many tasks such as object recognition and localization. The problem of matching can be defined as establishing a mapping between features in one image and similar features in another image. Representing image feature configurations as attributed graphs has long been a powerful paradigm in computer vision, in which the nodes represent the abstraction of image features and the edges capture spatial relations among features. When feature configurations are represented as graphs, the feature matching problem translates into establishing correspondence (or isomorphism) between the associated graphs.

The problem of computing the similarity between pairs of objects using their graph representations has been the subject of extensive studies in the computer vision and pattern recognition communities for over two decades. Most of the work on the matching of attributed graphs has focused on the problem of establishing injective correspondences between the vertices of two graphs. Haralick and Shapiro [21] proposed one of the earliest such frameworks for subgraph isomorphism based on inexact matching of weighted primitives (weighted attributes and weighted relation

tuples) associated with each graph. Their algorithm inspired inexact injective graph matching algorithms based on tree search techniques with backtracking [24]. Heuristic-based graph traversal [26], graph editing [5,25], probabilistic relaxation [12], linear programming [3], and optimization techniques [4] exemplify inexact graph matching algorithms. Recently, heuristics are used in some frameworks to speed up the search time of the algorithms in this class (e.g., [14,13]).

The assumption of one-to-one correspondence is a very restrictive one, for it assumes that the primitive features in the two graphs agree in all aspects of their abstraction. Unfortunately, there are a variety of conditions that may lead to graphs that encode similar image feature configurations yet fail to match in a one-to-one fashion. This can be due to noisy feature detection, segmentation errors, or failure in detection of features at their appropriate scales. This, in turn, implies that a single feature (node) in one graph (image) may map to a collection or cluster of features (nodes) in another graph. The limitation of exact (or, one-to-one) graph matching is depicted in Fig. 1, where the silhouettes of two shapes and their undirected shock graphs [22] are shown at the top and bottom rows, respectively. Although the elephant silhouettes are similar, no graph or subgraph isomorphism exist between their graphs. Some parts that exist in one graph do not appear in the other. Moreover, some parts such as rear legs of the left silhouette are shown in a different appearance on the right,

* Corresponding author.

E-mail address: mfdemirci@etu.edu.tr (M.F. Demirci).

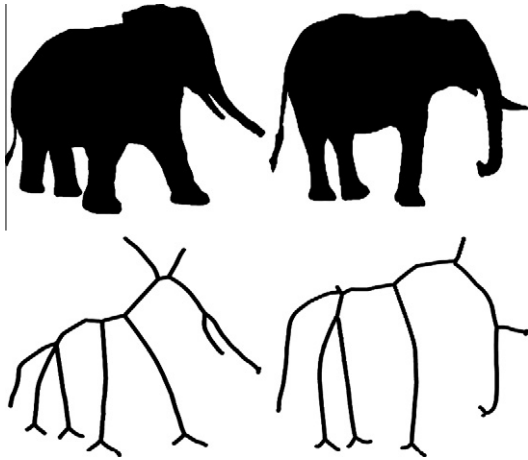


Fig. 1. The need for many-to-many matching. Although the silhouettes of two elephants are similar, a one-to-one matching does not exist between their shock graphs. Some parts which exist in one silhouette do not appear in the other. A robust algorithm should establish many-to-many correspondence between image features.

resulting in a different structure in the graph. As a result, a robust matching algorithm should be able to establish many-to-many correspondence between image features.

In previous work [8], we presented an inexact graph matching algorithm that establishes many-to-many correspondences among the vertices of two graphs by first embedding them in a metric space as two distributions and then solving a geometric transportation problem to align the two sets. More specifically, the framework maps vertices of the input graphs into the same geometric space through a graph embedding technique based on spherical coding. Throughout the embedding, attributed vertices in each graph map to weighted points, and shortest path distances between the vertices are mapped to the Euclidean distances between their corresponding points.

The embedding can be used only if the input graphs are trees. If the input graphs are not trees, i.e., they contain cycles, then a preprocessing step for constructing the metric tree approximation of the input graph is necessary. Next, the problem of many-to-many graph matching will be reformulated as that of many-to-many point matching. The approach for solving the many-to-many matching of the weighted point sets is based on using the Earth Mover's Distance (EMD) [19] algorithm.

The approach provides favorable matching results, but suffers from a major drawback. Namely, the graph embedding used in this work results in distortion, i.e., shortest-path distances between the vertices are not always equal to the Euclidean distances between their corresponding points. Although the distortion is theoretically bounded, this reduces the representative power of graphs and thus results in an approximate representation in the geometric space.

In this paper, we overcome the problem of approximate graph representation in the geometric space through an isometric embedding technique. Drawing on an important theorem from graph theory, we ensure that the distances in the input graphs (trees) are preserved exactly in the geometric space under the l_1 norm [17], which is of particular interest to many applications such as music retrieval and approximate matching. Armed with a distortion-free embedding of an input graph's structure, many-to-many graph matching is reduced to the problem of matching point sets, for which we use the EMD algorithm. We show that the many-to-many point matching that realizes the minimum EMD corresponds to the desired many-to-many matching between vertices of the

original graphs. Moreover, experimental evaluation of the proposed work demonstrates that the framework is a more efficient, more effective and more stable approach to many-to-many graph matching over the previous algorithm. Our preliminary work on many-to-many matching under l_1 is reported in [7].

The rest of the paper is organized as follows. After providing a brief overview of the previous many-to-many matching algorithm in Section 2, we describe the extension of this algorithm and analyze its complexity in Section 3. Performing an isometric embedding not only improves the efficacy of the framework, it also reduces the overall complexity using recent developments in EMD under the l_1 norm. We evaluate and compare the proposed approach and previous work on a view-based recognition domain using two different datasets in Section 4. Finally, we close the paper with a conclusion in Section 5.

2. Overview of the many-to-many matching algorithm

Our previous many-to-many matching algorithm starts by representing input graphs as trees. The problem of fitting a distance matrix by a tree metric is known as the *Numerical Taxonomy* problem. Since construction of the numerical taxonomy tree is NP-hard under l_1 and l_2 norms and it is an open problem for general distance metrics under l_∞ , an approximation framework proposed by Agarwala et al. is used [1]. Given a graph, the objective of its tree construction procedure is to minimize the distance given by $\|T - D\|_k$, where T is the tree metric, D is the metric defined on the vertices of the input graphs using shortest path distances, and $k \geq 1$. Agarwala et al.'s procedure allows us to use the many-to-many matching technique for any type of graph. However, if we assume that database silhouettes can be represented directly by trees, this process is not needed. In this paper, we compute a tree representation for each silhouette in the database without employing Agarwala et al.'s procedure. The details of the tree representation algorithm are defined in Section 4.

Given two trees, the many-to-many matching algorithm embeds them into a geometric space of prescribed dimensionality. This step is performed by a graph embedding technique using spherical coding. The goal of the graph embedding is to map a graph's structure to a set of vectors in a low-dimensional space. This mapping simplifies the original graph representation and retains important information about both local (neighborhood) as well as global graph structure. In addition, this mapping is stable with respect to noise in the graph structure.

Having defined a low-dimensional, robust vector representation of an input graph's structure, many-to-many graph matching is reduced to the much simpler problem of matching weighted distributions of points in a geometric space using a distribution-based dissimilarity measure, known as EMD. The EMD algorithm evaluates the dissimilarity between two multi-dimensional distributions. The main advantage of using EMD is that it subsumes many histogram distances and permits partial matches in a natural way. This important property allows the similarity measure to deal with uneven clusters and noisy datasets.

Computing the EMD is based on the well-known *transportation problem* [2], which is formulated as linear programming. Assume that the elements in the first distribution indicate supplies and the elements in the second distribution indicate demands at their positions. The EMD then computes the minimum amount of work required to transform one distribution into the other. Formally, let $P = \{(p_1, w_{p_1}), \dots, (p_n, w_{p_n})\}$ be the first distribution of size n and $Q = \{(q_1, w_{q_1}), \dots, (q_m, w_{q_m})\}$ be the second distribution of size m , where p_i (or q_i) is the position of the i th element and w_{p_i} (or w_{q_i}) is its weight, and let d_{ij} denote the ground distance between points p_i and q_j . The objective function of this problem is to find a flow

matrix $F = [f_{ij}]$, with f_{ij} being the amount of flow between p_i and q_j , which minimizes the overall cost:

$$\text{EMD}(P, Q) = \sum_{i=1}^n \sum_{j=1}^m f_{ij} d_{ij}$$

subject to $\sum_{i=1}^n f_{ij} \leq w_{p_i}$, $\sum_{j=1}^m f_{ij} \leq w_{q_j}$, $\sum_{i=1}^n \sum_{j=1}^m f_{ij} = \min(\sum_{i=1}^n w_{p_i}, \sum_{j=1}^m w_{q_j})$, and $f_{ij} \geq 0$. An efficient solution to the EMD problem plays critical role in the effectiveness of many-to-many applications, e.g., image retrieval and shape matching in computer vision. Transportation-simplex methods, interior-point algorithms, and minimum cost network flow techniques are some examples of approaches for solving the transportation problem efficiently. In this paper, we use the transportation-simplex method [11].

The solution of the EMD yields three different types of correspondences: one-to-one, one-to-many, and many-to-one. While one-to-one correspondence is usually obtained when the weight of each point in the input distributions is uniform, one-to-many and many-to-one correspondences can result from point distributions with distinct weights. In a one-to-many correspondence, a set of points $\{(q_1, w_{q_1}), \dots, (q_k, w_{q_k})\} \in Q$ receive flow from a common point $(p', w_{p'}) \in P$ such that $\sum_{i=1}^k f_{p'i} \leq w_{p'}$. Similarly, in a many-to-one correspondence, a set of points $\{(p_1, w_{p_1}), \dots, (p_l, w_{p_l})\} \in P$ send flow to a single point $(q', w_{q'}) \in Q$ such that $\sum_{i=1}^l f_{i q'} \leq w_{q'}$. Overall, the solution provided by the EMD leads a many-to-many correspondence.

An extension of the original EMD approach matches point sets that are non-rigidly embedded into the Euclidean space by allowing sets to undergo transformations [6]. Assuming that a transformation is applied to the second distribution, distances d_{ij}^t are defined as $d_{ij}^t = d(p_i, T(q_j))$ and the objective function becomes $\text{EMD}(P, Q) = \sum_{i=1}^n \sum_{j=1}^m f_{ij} d_{ij}^t$. The minimal value of the objective function $\text{EMD}(P, Q, T)$ defines the Earth Mover's Distance between the two distributions that are allowed to undergo a transformation.

Applying the many-to-many point correspondences produced by the EMD back to the original graphs yields the many-to-many graph matching. The solution of the EMD also represents the distance between the point sets. This distance is then used as a dissimilarity score between the original graphs. Thus, given a query and a database, the database graph which most resembles the query graph can be retrieved by computing the dissimilarity score between the query and each database graph, and choosing the closest graph. An overview of the many-to-many matching algorithm is given in Fig. 2.

3. Embedding trees under the l_1 norm and many-to-many point matching

The many-to-many matching approach outlined in the previous section transforms a combinatorial inexact graph matching problem into an instance of the geometric point-set matching problem. Since the efficacy of the algorithm depends on the distribution of edge weights in the original graphs, i.e., the weight of an edge is inversely proportional to the chance that the endpoints (vertices) of the edge map to the same conic section of spherical their encoding. Preserving the vertex proximity when embedding them in Euclidean space will allow them to receive flow from a common source during the EMD algorithm. However, embedding a tree into Euclidean space under the l_2 norm may introduce considerable distortion in the original edge weights. Since distances between the vertices of a tree are not represented accurately by those of the points in the Euclidean space, the final many-to-many matching obtained by the EMD algorithm may result in both undesired vertex correspondences and dissimilarity values between the trees. The following result, due to Gupta [10], provides an upper-bound on the

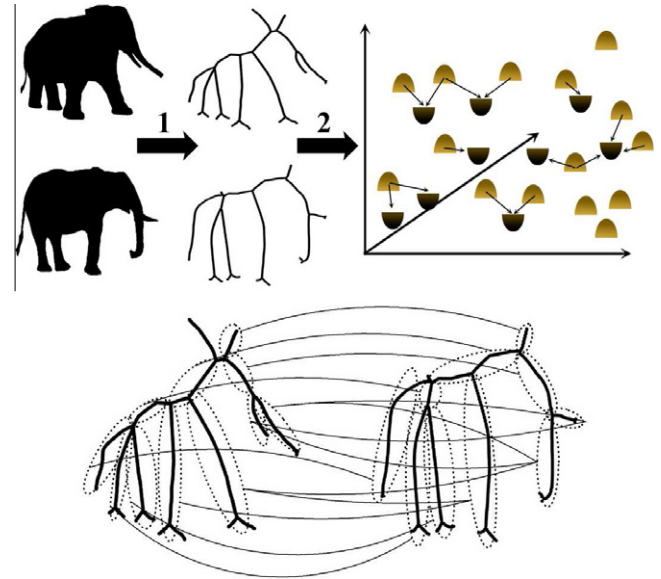


Fig. 2. Overview of the many-to-many matching algorithm. After representing two silhouettes as shock graphs (transition 1), the graphs are embedded into geometric spaces of the same dimensionality (transition 2). To compute the matching between the two distributions, the Earth Mover's Distance (EMD) algorithm is used. The bottom part illustrates the many-to-many correspondences between the vertices of the input graphs, defined by the computed flows from the EMD. Each dashed ellipsoid represents a set of vertices from the original graph.

magnitude of distortion in embedding a tree into a geometric space under l_2 .

Theorem 1. For a tree T with $\ell(T)$ leaves, there exists an embedding $\phi : V(T) \rightarrow \mathbb{R}_d^d$ with distortion:

$$D(\phi) \leq O\left(\ell^{1/(d-1)}(\min\{\log \ell(T), d\})^{1/2}\right) \quad (1)$$

where d is the dimension of the geometric space.

In practice, for large trees, embedding algorithms under l_2 may result in distortion. In fact, experimental results of Demirci et al. [8] indicate that spherical embedding under l_2 can cause an average of 17% distortion in a 100-dimensional space.

In this paper, we will overcome this problem by embedding trees under l_1 . The idea is to use an isometric embedding under the l_1 norm [10] that can ensure no distortion is introduced when transforming the many-to-many matching problem from the graph domain to geometric space. In a d -dimensional metric space, the distance U_1 , between two points $X = [x_1, x_2, \dots, x_d]^t$ and $Y = [y_1, y_2, \dots, y_d]^t$ can be computed under the l_1 norm as

$$U_1(X, Y) = \sum_{k=1}^d |x_k - y_k|. \quad (2)$$

Our proposed embedding uses the caterpillar decomposition that captures the topological structure of the tree and is defined as the collection of edge-disjoint (sub)root-leaf paths. This concept is illustrated in a sample tree shown in Fig. 3. The three paths between a and c , a and g , a and m are called level 1 paths and represent the first three paths in the caterpillar decomposition. If we remove these three level 1 paths from the tree, we are left with three edge-disjoint paths. These are the paths between e and f , i and j , k and l , called level 2 paths, which represent the other three paths in the caterpillar decomposition. If removing the level 2 paths had left additional connected components, the process would be repeated until all the edges in the tree had been removed. The union of the paths is called the caterpillar decomposition, denoted by \mathfrak{P} . The total number of paths in \mathfrak{P} specifies the dimensionality of the

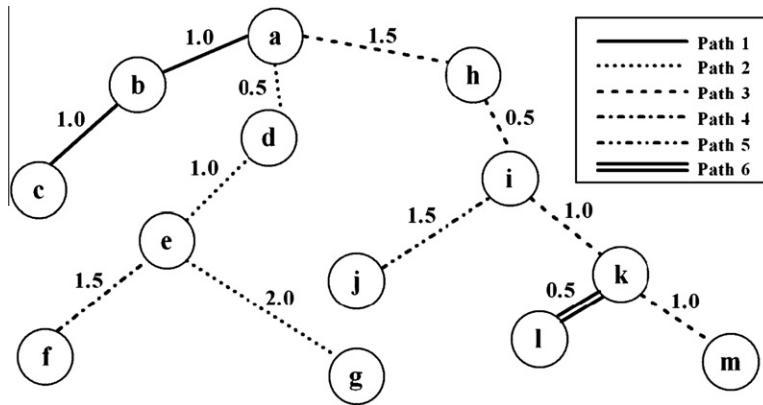


Fig. 3. Edge-disjoint paths extracted from the caterpillar decomposition of a tree.

geometric space into which the graph is embedded. For a detailed description of the Caterpillar decomposition and its properties and computation, see Matoušek [18].

To compute the coordinates of vertex v in the geometric space, we first find the unique path $P(v)$ between v and the root r . Assume that the first segment of $P(v)$ of weight ℓ_1 follows some path $P^1 \in \mathfrak{P}$, the second segment of weight ℓ_2 follows a path $P^2 \in \mathfrak{P}$, and the last segment of weight ℓ_a follows a path $P^a \in \mathfrak{P}$. Let the sequences $\langle P^1, \dots, P^a \rangle$ and $\langle \ell_1, \dots, \ell_a \rangle$ be the decomposition sequence and the weight sequence of $P(v)$, respectively. Since each path in \mathfrak{P} corresponds to a coordinate axis, the following process is used to find the coordinates of v : if the decomposition sequence of $P(v)$ consists of a path $P^i \in \mathfrak{P}$, its corresponding coordinate will have a value of ℓ_i as defined in the weight sequence. Otherwise, the corresponding coordinate will have a value of 0. It is easy to see that the embedding obtained through this procedure is isometric under l_1 . To illustrate this procedure, we turn back to Fig. 3 in which the tree is embedded into a 6-dimensional space. For example, to compute the coordinates of vertex f , observe that the path between a and f

consists of one level 1 path (between a and e) of weight 1.5 and one level 2 path (between e and f) of weight 1.5. Since these paths correspond to the 2nd and 4th paths in the caterpillar decomposition of the tree, only the 2nd and 4th coordinates of the point representing f will be non-zero. Thus, the coordinates of vertex f in the geometric space are $(0, 1.5, 0, 1.5, 0, 0)$. Although the caterpillar decomposition \mathfrak{P} is not unique, the resulting embeddings are all isometric under l_1 . However, to be consistent in our embedding procedure, the order in which the paths in the caterpillar decomposition are selected is defined by their weights. After computing the coordinates of each vertex in the tree, the distance between any pair of vertices in the tree is preserved by the l_1 distance between their corresponding coordinates. Since the geometric properties of a tree are represented exactly under the l_1 norm, this embedding is distortion-free.

It is important to note that embeddings produced by the above algorithm can be of different dimensions. Therefore, in order to match the two embeddings, we must first perform a registration step, whose objective is to project the two distributions into the



Fig. 4. Sample silhouettes from the first dataset (db1).

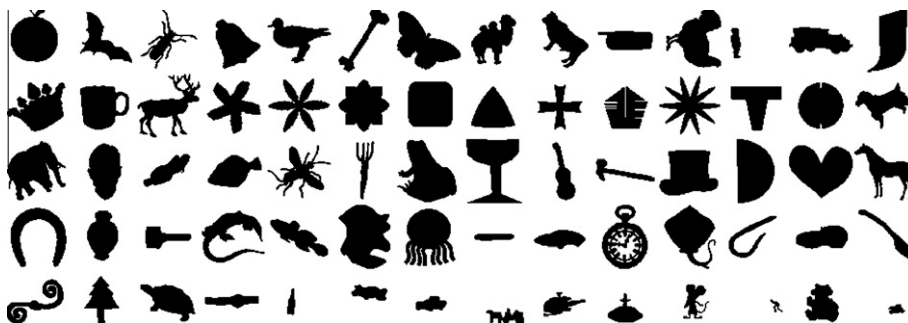


Fig. 5. Representative silhouettes of each class from the MPEG-7 database.

same normed space. As done in our previous work [9], we could, for instance, project the distributions onto the first K right singular vectors of their covariance matrices. This technique is based on Principal Component Analysis and retains the maximum information about the original vectors among all projections onto subspaces of dimension K . Although this method equalizes the dimensions of the two distributions while losing minimal information, it still introduces distortion in the geometric space. Since our objective is to perform a distortion-free embedding in the proposed frame-

work, this technique is not suitable for our purpose. Our solution for equalizing the dimensions of the distributions is simply based on padding zeros to the lower dimensional distribution. Since this increases the complexity of the EMD algorithm, we seek a more efficient implementation of this technique in our framework.

Having isometrically embedded the trees into the same dimension, we can now proceed with finding the matching between the points. As mentioned earlier, this step is performed using the EMD algorithm. To generate the many-to-many matching efficiently

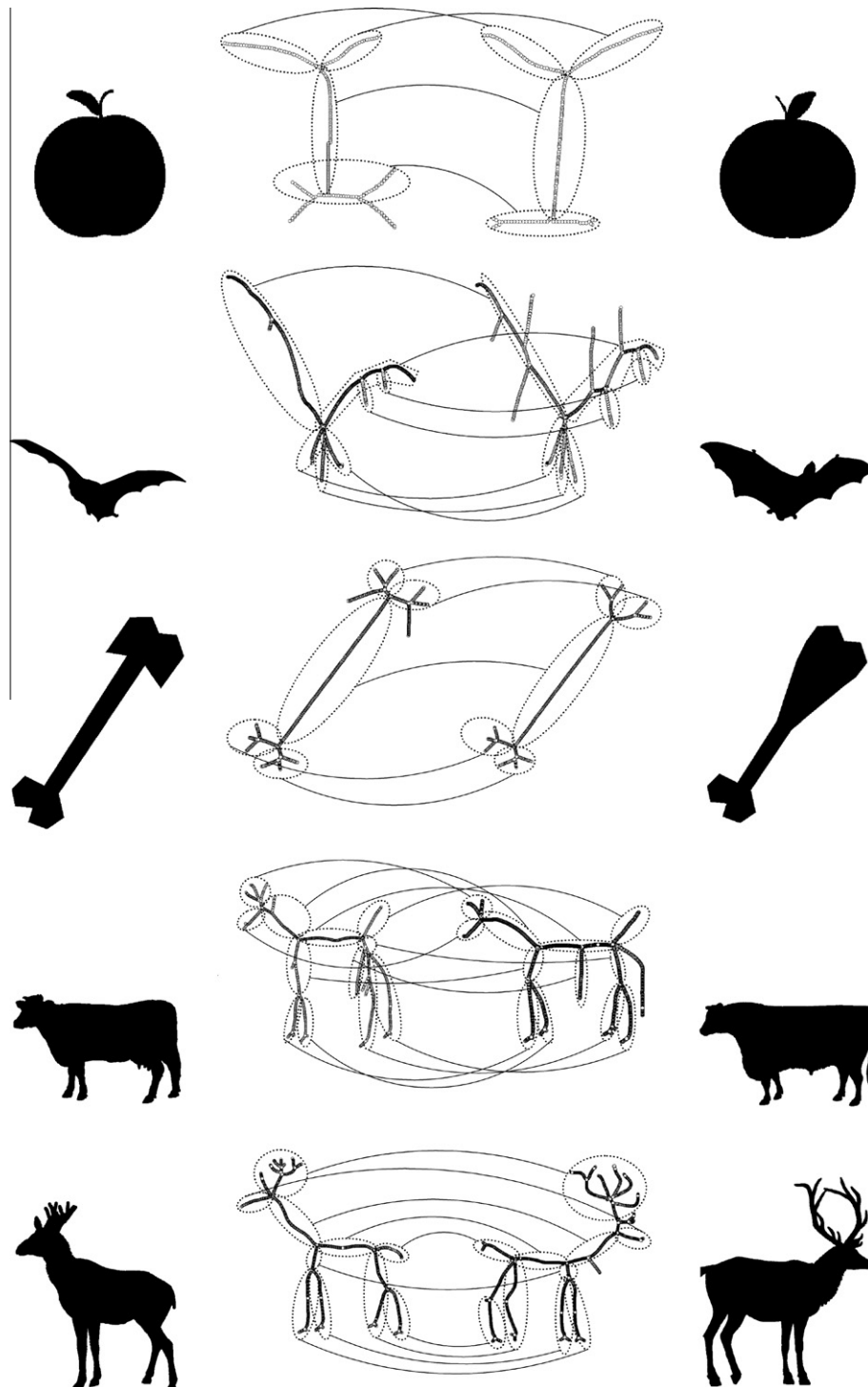


Fig. 6. Matching results for some MPEG-7 shapes. The skeleton groups shown inside corresponding ellipses are matched many-to-many.

under l_1 , we use the EMD- l_1 approach presented in [16], which has a simplified structure and improved complexity in comparison to the original EMD formulation. The authors also describe an efficient tree-based algorithm to solve the EMD- l_1 . In addition, they showed that the EMD- l_1 is equivalent to the original EMD with l_1 ground distance without approximation.

3.1. Complexity

The proposed algorithm consists of several components. Thus, we start by stating the complexity of each component. Computing the tree for a given graph takes $O(|V|^2)$ (see [1] for details). Performing an isometric embedding of trees into the geometric space under l_1 can be done in linear time using depth first search as shown before. The EMD is formulated as a linear programming problem and can be solved using a network flow algorithm in $O(|V|^3)$. However, using EMD- l_1 , the complexity of the EMD is reduced to $O(|V|^2)$. Finally, mapping the EMD solution back to the graph is $O(|V|)$. Thus, the overall complexity of the proposed approach is bounded by $O(|V|^2)$. This shows that by employing a distortion-free embedding and using the EMD- l_1 approach, the proposed framework has a better time complexity than the previous work (i.e., $O(|V|^{7/6}|E|)$).

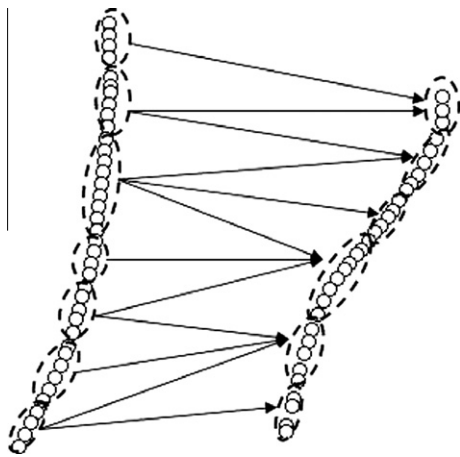


Fig. 7. The actual many-to-many correspondences between skeleton groups within the upper left ellipses of the two apple images shown in Fig. 6. The skeleton groups are rotated for better visibility.

4. Experiments

In this section, we present the experimental evaluation of the proposed method and its comparison with the previous work [8] in the context of an object recognition experiment using two different datasets, which both comprise 2D shapes of 3D objects. The first dataset (db1) consists of 1620 silhouettes of 9 objects, with 180 views for each; example silhouettes are shown in Fig. 4. The second database is the MPEG-7 CE-Shape-1 (Part B) database, consisting of 1400 shapes clustered into 70 classes with 20 shapes per class. Fig. 5 represents sample silhouettes of each class from this dataset. Each silhouette in each dataset is represented by an undirected shock tree, whose nodes represent shock points [22] and whose edges connect adjacent shock points. Each vertex v in the tree is labeled by a triplet (x,y,r) , where the pair (x,y) denotes the Euclidean coordinates of the corresponding shock point p and r is the radius of the maximal bi-tangent circle centered at p . Each shock point becomes a node in the shock graph. Each pair of shock points will be connected by an edge whose weight reflects the Euclidean distance between the corresponding shock points. The graph is converted into a tree by computing its minimum spanning tree. Thus, tree nodes correspond to shock points, and tree edges connect nearby shock points. We choose the root of the tree to be the node that minimizes the sum of the tree-based shortest path distances to all other nodes. The root choice may be sensitive to occlusion and noise. We plan to evaluate other strategies such as selecting the node with maximum degree as the root in the future.

As an illustration of our approach, in Fig. 6 we present some examples of the many-to-many matchings obtained from our algorithm for some object pairs selected from the MPEG-7 database. In this figure, skeleton groups corresponding to each other are placed inside ellipses and these ellipses are connected by a curve. The corresponding skeleton groups are matched many-to-many. One may notice that the correspondences are intuitive in all examples. Fig. 7 displays the actual many-to-many correspondences between skeleton groups within the upper left ellipses of the two apple images (first row of Fig. 6).

To provide a more comprehensive evaluation of the framework, we conducted the following experiment for each dataset. We first use each database graph as a query (with replacement) to the remaining database. Then, we compute the distance between each query and each of the remaining database graphs using our proposed algorithm. Ideally, given a query view of an object, the matching algorithm should return another view of the same object.

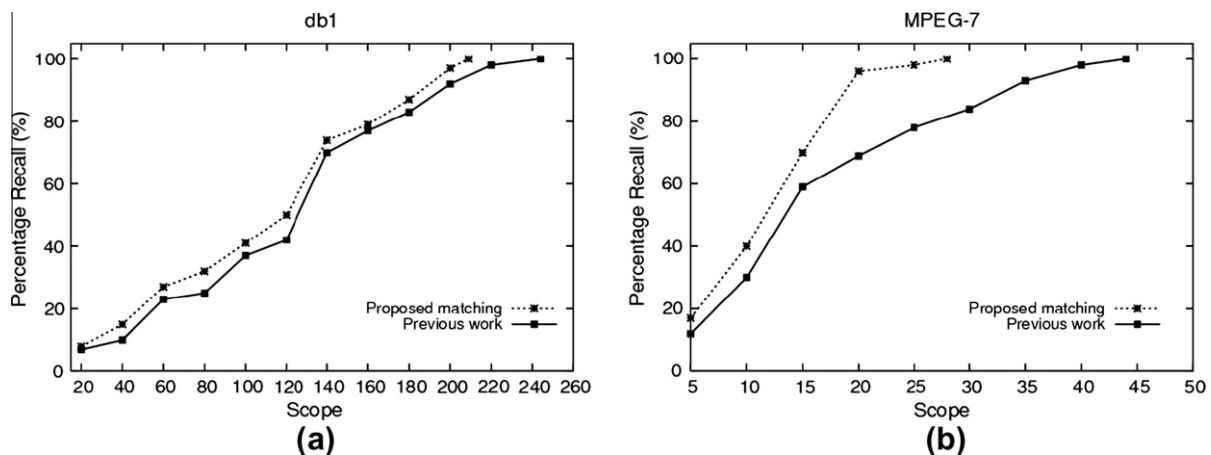


Fig. 8. Percentage recall values of the proposed and previous recognition techniques for db1 (a) and the MPEG-7 dataset (b). Observe that the proposed algorithm is superior for each percentage recall value in both datasets.

This is classified as a correct object recognition. According to the results, our framework and the previous work obtain 97.5% and 95.3% object recognition rates, respectively, for db1. Repeating the same experiment for the MPEG-7 dataset, our framework achieves 93.8%, while the previous work results in 89.6% object recognition rate.

In the second experiment, we compute the total number of retrieved images that is necessary to retrieve the entire query class. After computing the similarity values between every database pair, we look at the top matches to see how many of the within-category shapes belong to the same class as the query. Ideally, if an object has k shapes in the database, the top $k - 1$ entries should belong to the same class as the query. The results show that the first 209 closest database objects always contain all the views belonging to the query class for db1 using the proposed framework. This quantity was 28 for MPEG-7 database. Performing the same experiments using the previous work yields 244 for db1 and 44 for MPEG-7. These values indicate that the recall in each dataset is 100% if the scope is set to the first 12.9% and 2% of the closest database objects using our framework, and 15% and 3.1% using the previous approach. In Fig. 8, we show percentage recall values for various scopes for each dataset using both many-to-many matching techniques.

The results clearly demonstrate the improved performance offered by the isometric embedding technique, at a reduced compu-

tational cost. To demonstrate the framework's robustness, we performed four perturbation studies on both the databases. The experiments are identical to the ones described above, except that the query was perturbed by adding/deleting 5%, 10%, 15%, and 20% of its nodes (and their adjoining edges). The results are shown in Table 1, and reveal that the error rates increase gracefully as a function of increased perturbation. Although this is not a true occlusion experiment, which would require that a subtree is replaced with an occluder subtree, the results demonstrate the framework's ability to match local structure, which is essential for handling occlusion.

We should note that upon investigation of our results, we found that most mismatches in MPEG-7 were between similar classes. Considering that there are seven different tool classes and several 4-legged animals in MPEG-7, our results should be considered as worst-case. Despite being different objects, their shapes and thus their skeletons are similar. Grouping these similar classes into the same category will indeed improve our recognition rates.

Having demonstrated the improved performance of the proposed technique over the previous many-to-many matching algorithm and its robustness under perturbation, we compare our approach to a leading graph edit-distance approach [20], a generative model approach [23], and an inner distance approach [15] next. The database used for this experiment consists of 99 instances from nine classes provided by Kimia's group (Fig. 9). Similar to the retrieval experiments presented above, each shape is matched against all other shapes in the database. Recognition scores are given as the number of k th closest matches that fall into the correct class, where $k = 1, \dots, 10$. The best possible score in each match is 99. The results are presented in Table 2 and reveal that our approach performs comparably to the best approaches reported on this dataset. The proposed framework is the second best approach based on the total number of mismatches in top 10 retrievals, which is recorded as 102, 34, 32, 26, and 29 for generative models, graph edit-distance, inner distance (IDSC + DP), inner distance (MDS + SC + DP) and the proposed algorithm, respectively.

Table 1

Recognition rate as a function of increasing perturbation. The baseline recognition rate (with no perturbation) is 97.5% for db1 and 93.8% for MPEG-7.

Perturbation	5%	10%	15%	20%
Recognition rate db1	95.13%	90.73%	84.25%	78.72%
Recognition rate MPEG-7	90.32%	85.28%	79.15%	74.41%

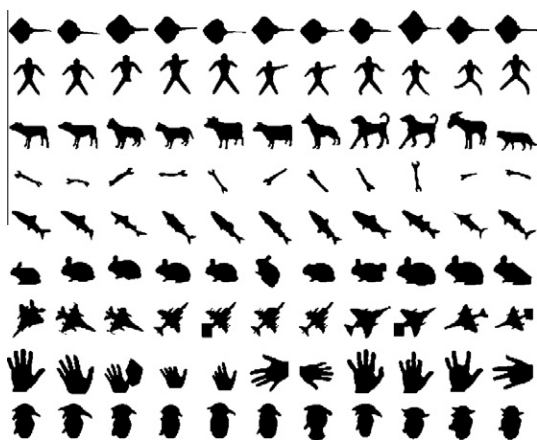


Fig. 9. Kimia's dataset [20] consisting of 99 silhouettes in nine classes.

5. Conclusions

Matching object features many-to-many is a critical process for object recognition and classification. One-to-one matching algorithms cannot handle segmentation/articulation errors or scale difference, which may exist between features of similar objects. This paper computes the dissimilarity between object pairs represented as trees by computing the many-to-many matching between their vertices. The proposed framework embeds the input trees isometrically into a geometric space under the l_1 norm. Unlike previous work, the embedding approach used in the proposed framework is distortion-free. The embedded points are efficiently matched many-to-many using a recent development in the efficient Earth Movers Distance (EMD) algorithm under l_1 . The resulting flows specify the many-to-many vertex correspondences between the input trees. We have successfully evaluated the proposed technique on three image databases and include a set of structural perturba-

Table 2

The number of top 1–top 10 closest matches that fall into the correct class are shown for several algorithms on Kimia's dataset. The best possible score in each match is 99. Observe that our approach performs comparably to the best approaches.

Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Generative models [23]	99	97	99	98	96	96	94	83	75	48
Graph edit-distance [20]	99	99	99	98	98	97	96	95	93	82
Inner distance (IDSC + DP) [15]	99	99	99	98	98	97	97	98	94	79
Inner distance (MDS + SC + DP) [15]	99	98	98	98	97	99	97	96	97	85
Proposed method	99	99	98	99	98	97	98	97	92	84

tion experiments that establish the algorithm's robustness to noise. Moreover, we have shown that the approach compares favorably to previous approaches on these datasets. Studying different isometric embedding techniques under various norms and employing other distribution-based matching algorithms in our framework are our future plans.

Acknowledgments

The work of Fatih Demirci is supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) Grant 109E183. Ali Shokoufandeh gratefully acknowledges the support from the National Science Foundation Grant #0803670 under the IIS Division and Office of Naval Research Grant ONR-N000140410363. Sven Dickinson acknowledges the support of NSERC Canada.

References

- [1] R. Agarwala, V. Bafna, M. Farach, M. Paterson, M. Thorup, On the approximability of numerical taxonomy (fitting distances by tree metrics), *SIAM Journal on Computing* 28 (2) (1999) 1073G–1085G.
- [2] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [3] H.A. Almohamad, S.O. Duffuaa, A linear programming approach for the weighted graph matching problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (5) (1993) 522–525.
- [4] E. Bengoetxea, P. Larranaga, I. Bloch, A. Perchant, C. Boeres, Inexact graph matching by means of estimation of distribution algorithms, *Pattern Recognition* 35 (2002) 2867–2880. 14.
- [5] H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (9) (1999) 917–922.
- [6] S.D. Cohen, L.J. Guibas, The earth mover's distance under transformation sets, in: *Proceedings, 7th International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1076–1083.
- [7] F. Demirci, Y. Osmanlioglu, Many-to-many matching under the l_1 norm, in: *Proceedings of the 15th International Conference on Image Analysis and Processing*, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 787–796.
- [8] F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, S. Dickinson, Object recognition as many-to-many feature matching, *International Journal of Computer Vision* 69 (2) (2006) 203–222.
- [9] F. Demirci, A. Shokoufandeh, Y. Keselman, S. Dickinson, L. Bretzner, Many-to-many matching of scale-space feature hierarchies using metric embedding, in: *Proceedings of the 4th International Conference on Scale Space Methods in Computer Vision*, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 17–32.
- [10] A. Gupta, Embedding tree metrics into low-dimensional euclidean spaces, *Discrete & Computational Geometry* 24 (1) (2000) 105–116.
- [11] F. Hillier, G. Lieberman, *Introduction to Mathematical Programming*, McGraw-Hill, New York, NY, 1990.
- [12] J. Kittler, W.J. Christmas, M. Petrou, Probabilistic relaxation for matching of symbolic structures, in: *Workshop Structural and Syntactic Pattern Recognition*, Bern, Switzerland, 1992, pp. 471–480.
- [13] C. Sansone, L. Cordella, P. Foggia, M. Vento, A (sub)graph isomorphism algorithm for matching large graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (10) (2004) 1367–1372.
- [14] J. Larrosa, G. Valiente, Constraint satisfaction algorithms for graph pattern matching, *Mathematical Structures in Computer Science* 12 (4) (2002) 403–422.
- [15] H. Ling, D. Jacobs, Shape classification using the inner-distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2) (2007) 286–299.
- [16] H. Ling, K. Okada, An efficient earth mover's distance algorithm for robust histogram comparison, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (5) (2007) 840–853.
- [17] N. Linial, E. London, Y. Rabinovich, The geometry of graphs and some of its algorithmic applications, in: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1994, IEEE Computer Society, pp. 577–591.
- [18] J. Matoušek, On embedding trees into uniformly convex Banach spaces, *Israel Journal of Mathematics* 237 (1999) 221–237.
- [19] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, *International Journal of Computer Vision* 40 (2) (2000) 99–121.
- [20] T. Sebastian, P. Klein, B. Kimia, Recognition of shapes by editing their shock graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (5) (2004) 550–571.
- [21] L.G. Shapiro, R.M. Haralick, Structural descriptions and inexact matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3 (5) (1981) 504–519.
- [22] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, *International Journal of Computer Vision* 35 (1) (1999) 13–32.
- [23] Z. Tu, A. Yuille, Shape matching and recognition: using generative models and informative features, in: *European Conference on Computer Vision*, vol III, 2004, pp. 195–209.
- [24] J.R. Ullmann, An algorithm for subgraph isomorphism, *Journal of the Association for Computing Machinery* 23 (1) (1976) 31–42.
- [25] Y.K. Wang, K.C. Fan, J.T. Horn, Genetic-based search for error-correcting graph isomorphism, *IEEE Transactions on Systems, Man, and Cybernetics* 27 (1997).
- [26] M.L. Williams, R.C. Wilson, E.R. Hancock, Deterministic search for relational graph matching, *Pattern Recognition* 32 (7) (1999) 1255–1271.