An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison

Haibin Ling
Center for Automation Research, Computer Science Department
University of Maryland,
College Park, Maryland, 20770, USA
hbling@umiacs.umd.edu

Kazunori Okada
Imaging and Visualization Department
Siemens Corporate Research, Inc.
755 College Rd. E. Princeton, New Jersey, 08540, USA
kazunori.okada@siemens.com

A preliminary version of this paper appeared in ECCV'06 [28]

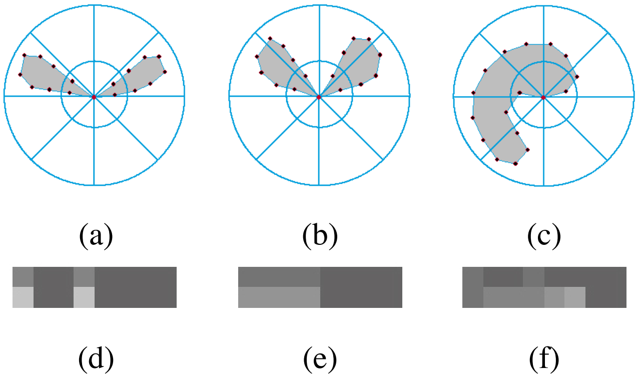
Abstract

We propose EMD-L_1: a fast and exact algorithm for computing the Earth Mover's Distance (EMD) between a pair of histograms. The efficiency of the new algorithm enables its application to problems that were previously prohibitive due to high time complexities. The proposed EMD-L_1 significantly simplifies the original linear programming formulation of EMD. Exploiting the L_1 metric structure, the number of unknown variables in EMD-L_1 is reduced to $O(N)$ from $O(N^2)$ of the original EMD for a histogram with N bins. In addition, the number of constraints is reduced by half and the objective function of the linear program is simplified. Formally without any approximation, we prove that the EMD-L_1 formulation is equivalent to the original EMD with a L_1 ground distance. To perform the EMD-L_1 computation, we propose an efficient tree-based algorithm, Tree-EMD. Tree-EMD exploits the fact that a basic feasible solution of the simplex algorithm-based solver forms a spanning tree when we interpret EMD-L_1 as a network flow optimization problem. We empirically show that this new algorithm has average time complexity of $O(N^2)$, which significantly improves the best reported super-cubic complexity of the original EMD. The accuracy of the proposed methods is evaluated by experiments for two computation-intensive problems: shape recognition and interest point matching using multi-dimensional histogram-based local features. For shape recognition, EMD-L_1 is applied to compare shape contexts on the widely tested MPEG7 shape dataset, as well as an articulated shape dataset. For interest point matching, SIFT, shape context and spin image are tested on both synthetic and real image pairs with large geometrical deformation, illumination change and heavy intensity noise. The results demonstrate that our EMD-L_1-based solutions outperform previously reported state-of-the-art features and distance measures in solving the two tasks.

Index Terms Earth Mover's Distance, transportation problem, histogram-based descriptor, SIFT, shape context, spin image, shape matching, interest point matching

I. Introduction

Histogram-based local descriptors are ubiquitous tools in numerous computer vision tasks, such as shape matching [3], [32], [47], [48], [26], image retrieval [29], [21], [34], [31], [27], texture analysis [22], color analysis [42], [45], and 3D object recognition [18], [36], to name a few. For comparing these descriptors, it is common to apply *bin-to-bin* distance functions, including L_p distance, χ^2 statistics, KL divergence, and Jensen-Shannon (JS) divergence [25]. In applying these bin-to-bin functions, we often assume that the domain of the histograms are previously aligned. However, in practice, such an assumption can be violated due to various factors, such as shape deformation, non-linear lighting change, and heavy noise. The *Earth Mover's Distance*



Distance	$d(a,b)$	$d(b,c)$
L_1	1.0	0.875
L_2	0.3953	0.3644
χ^2	0.6667	0.6625
EMD	0.5	1.5625

(g)

Fig. 1. An example where bin-to-bin distances meet problems. (a),(b) and (c) show three shapes and log-polar bins on them. (d),(e) and (f) show the corresponding 2D histograms (shape context) of (a),(b) and (c) using the same 2D bins, respectively. The distances between (d) and (e) and the distances between (e) and (f) are summarized in table (g). All EMDs here use the L_1 ground distance.

(EMD) [42] is a *cross-bin* distance function that addresses this alignment problem. EMD defines the distance between two histograms as the solution of the *transportation problem* that is a special case of linear programming (LP). Beyond the color and texture signature application originally considered by Rubner et al. [42], we demonstrate in this article that EMD is useful for more general classes of histogram descriptors such as SIFT [29] and shape context [3].

Fig. 1 illustrates an example with the shape context, demonstrating the advantage of the cross-bin EMD over common bin-to-bin functions. The small articulation of two blobs between (a) and (b) causes a large change in their corresponding shape contexts as 2D histograms. EMD correctly describes the perceptual similarity of (a) and (b), while the three bin-to-bin distance functions, L_1, L_2 and χ^2, falsely state that (b) is more similar to (c) than to (a). Despite this favorable robustness property, EMD has seldom been applied to general histogram-based descriptors (especially local descriptors) to the best of our knowledge. The main reason lies in its expensive computational cost, which is larger than $O(N^3)$ (super-cubic[1]) for a histogram with N bins. Targeting this problem, we propose an efficient algorithm to compute EMD between histograms.

The contribution of this paper is twofold.

1) We propose a new fast algorithm, *EMD-L_1*, to compute EMD between histograms with L_1 ground distance. The formulation of EMD-L_1 is much simpler than the original EMD

[1] By super-cubic, we mean a complexity in $\Omega(N^3) \cap O(N^4)$.

formulation. It has only $O(N)$ unknown variables, which is significantly less than the $O(N^2)$ variables required in the original EMD. Furthermore, EMD-L_1 has only half the number of constraints and a more concise objective function. We prove that EMD-L_1 is formally equivalent to the original EMD with L_1 ground distance. As an optimization solver of EMD-L_1 computation, we designed an efficient tree-based algorithm. The new algorithm greatly improves the efficiency of the original transportation simplex algorithm as used in [42]. An empirical study demonstrates that the running time of EMD-L_1 quadratically increases with N, which is much faster than the previous super-cubic algorithm.

2) For the first time, EMD is successfully applied to compare histogram-based local descriptors. The speedup gained by EMD-L_1 enables us to compute EMD directly for multi-dimensional histograms without reducing the discriminability by introducing approximation. We tested the proposed approach in two tasks: shape matching and interest point matching. First, EMD-L_1 is applied to shape recognition tasks with the shape context [3] and the inner-distance shape context [26]. The experiments are conducted on two previously tested data sets, the widely tested MPEG7 shape data set and an articulated shape set. Our results show that EMD-L_1 outperforms all previously reported results. The second task is interest point matching for intensity images with three local descriptors, SIFT [29], shape context [3] and spin images [18], [22]. We experimented on both synthetic and real image pairs, under significant geometrical deformation, lighting change, and intensity noise. Again, EMD-L_1 demonstrates excellent performance. The results show that EMD-L_1 outperforms common metrics and works as accurately as the original, much slower, EMD with L_2 ground distance.

The rest of the paper is organized as follows. Sec. II discusses related works. In Sec. III, we review the original Earth Mover's Distance and present its formulation for histograms. Sec. IV introduces the proposed EMD-L_1, together with a formal proof of equivalence between EMD-L_1 and EMD with L_1 ground distance. Then a novel fast tree-based algorithm for computing EMD-L_1 is proposed in Sec. V. In Sec. VI, we report the results of experiments evaluating EMD-L_1 for shape matching using shape context and for interest point matching using SIFT, shape context and spin images. Finally, Sec. VII concludes this article and discusses our future work.

II. Related Work

Early works using a cross-bin matching cost for histogram comparison can be found in [44], [49] and [38]. Particularly, in Peleg et al. [38], images are modeled as sets of pebbles after normalization. The similarity between two images is the matching cost of two sets of pebbles based on the distances between them.

Adapted from previous work, the Earth Mover's Distance (EMD) is proposed by Rubner et al. [42] and Rubner and Tomasi [41] to compare distributions for image retrieval tasks. By modelling distribution comparison as a transportation problem [13] (a.k.a. the Monge-Kantorovich problem [39]), a specialized efficient linear programming algorithm, the *transportation simplex* (TS) algorithm [12] is proposed to solve the EMD. It is shown in [42] that TS has a super-cubic empirical time complexity. In [42], EMD is applied to signatures of distributions instead of directly to histograms. Signatures are abstracted representations of distributions and are usually clustered from histograms. This approach is very efficient and effective for distributions with sparse structures, e.g., the color histograms in the CIE-Lab space [42]. However, for histogram-based local descriptors that are not sparse in general, e.g. SIFT [29], EMD should be applied to histograms directly. In a typical setting to solve real vision problems, the number of required comparisons between these descriptors is very large, which forbids the use of the original TS algorithm. For example, to compare two images with 300 local features each, 90,000 comparisons are needed! Note that, although there is a fast exact EMD algorithm for 1D histograms [38], such a solution does not scale to higher dimensions - while most of the histogram-based local descriptors have two or three dimensions. In contrast, in this paper we extend it to general multi-dimensional histograms.

Since its initial proposal by Rubner et al. [42], EMD has attracted a large amount of research interest. Here we briefly summarize some examples. Cohen and Guibas [4] studied the problem of computing a transformation between distributions with minimum EMD. Levina and Bickel [24] proved that EMD is equivalent to the Mallows distance [30] when applied to probability distributions. Tan and Ngo [46] applied EMD for common pattern discovery using EMD's partial matching ability. Indyk and Thaper [16] proposed a fast approximation EMD algorithm and used it for image retrieval [16] and shape matching [7]. In addition, Holmes et al. [14], [15] touched on several areas explored in the paper, including EMD approximations in a Euclidean space for classes of derivative histograms and partial matching.

The fast algorithm proposed by Indyk and Thaper [16] is through embedding the EMD metric into a Euclidean space. The approach first embeds EMD between point sets into a L_1 space. This is done via some hierarchical distribution analysis. Then fast nearest neighbor retrieval is achieved via Locality-Sensitive Hashing (LSH). The EMD can then be approximated by the L_1 distance in the Euclidean space. Grauman and Darrell [7] extended the approach for fast contour matching. For this purpose, a shape is treated as a set of features on the contours, where each feature is treated as a point in the feature space. The time complexity of these algorithms are $O(Nd \log \Delta)$, where N is the number of points, d is the dimension of the feature space, and Δ is the diameter of the union of the two feature sets to be compared. These approaches are very efficient for retrieval tasks and global shape comparison [16], [7]. However, the approximation due to the embedding may sacrifice precision, reducing the discriminability of descriptors. As indicated in [16], the distortion upper bound is $O(\log \Delta)$ and empirical distortion is about 10%. In addition, these approaches focused on point set matching rather than the histogram comparison in which we are interested. Recently, Grauman and Darrell [8] proposed *pyramid matching kernel* (PMK) for feature set matching. PMK can be viewed as a further extension of the fast EMD embedding in that it also compare the two distributions in a hierarchical fashion. PMK also handles partial matching through histogram intersections [45].

In addition to EMD, other histogram dissimilarity measures and their performance evaluation can be found in [40]. Both bin-to-bin distances and cross-bin distances are discussed in [40], including the *quadratic form distance* [35], [10]. Quadratic form distance is another cross-bin distance. It allows comparison of histograms across different bin locations whose connectivity is heuristically determined by a quadratic form.

Unlike the above previous work, we focus on designing a distance metric for histogram-based local descriptors, which have attracted a lot of research interests recently [3], [32], [47], [48], [26], [29], [21], [34], [31]. Three representative examples are chosen in our experiments. First, the shape context, introduced by Belongie et al. [3], captures the distribution of landmark points. It is demonstrated to be very discriminative for shape matching. Some extensions of the shape context can be found in [32], [47], [48], [26]. The second is the scale invariant feature transform (SIFT) proposed by Lowe [29], which is a three-dimensional histogram measuring local gradient distributions. SIFT and its extensions are widely used for image matching and retrieval, e.g. [29], [21], [34], [31]. The third one is the spin image that basically computes the joint distribution of the intensity and distance of pixels around given interest points. It was first proposed by Johnson and Hebert [18] for 3D object recognition and later extended to a 2D texture descriptor by Lazebnik et al. [22]. A review of other descriptors and their performance evaluation can be found in [31]. Previously, these histogram-based local descriptors are compared by bin-to-bin metrics, especially the χ^2 distance and the L_p norms (e.g., Euclidean distance, Manhattan distance). In this paper, we will show that the proposed EMD comparison achieves better performance, especially for tasks involving large distortions including geometric deformation, illumination change and heavy intensity noise.

III. The Earth Mover's Distance (EMD)

A. The Original EMD between Signatures

The Earth Mover's Distance (EMD) is proposed by Rubner et al. [42] to measure the dissimilarity between signatures that are compact representations of distributions. A signature of size N is defined as a set $S = \{s_j = (w_j, m_j)\}_{j=1}^N$, where m_j is the position of the j-th element and w_j is its weight.

Given two signatures $P = \{(p_i, u_i)\}_{i=1}^m$ and $Q = \{(q_j, v_j)\}_{j=1}^n$ with size m, n respectively, the EMD between them is modeled as a solution to a transportation problem. Treat elements in P as "supplies" located at u_i and elements in Q as "demands" at v_j. Then p_i and q_j indicates the amount of supply and demand respectively. The EMD is defined as the minimum (normalized) work required for resolving the supply-demand transports, i.e.

$$EMD(P,Q) = \min_{F=\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}}$$

with the following constraints:

$$\sum_j f_{ij} \le p_i \ , \quad \sum_i f_{ij} \le q_j \ , \quad \sum_{i,j} f_{ij} = \min\{\sum_i p_i, \sum_j q_j\} \ , \quad f_{ij} \ge 0 \ ,$$

where $F = \{f_{ij}\}$ denotes a set of *flows*. Each flow f_{ij} represents the amount transported from the i-th supply to the j-th demand. We call d_{ij} the *ground distance* between the position u_i and v_j. Fig. 2 gives an example, where P has four elements and Q has three.

The transportation problem is a special case of linear programming (LP) problems. The constraint matrix in this case has a very sparse structure that enables an efficient algorithmic solution. One such efficient algorithm is the transportation simplex (TS) [42], [12]. Modified from

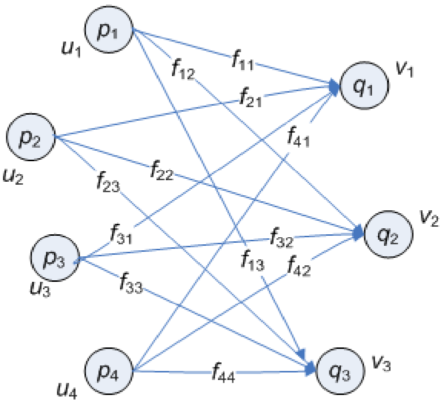


Fig. 2. EMD between two signatures ($m = 4$, $n = 3$) as a transportation problem.

the standard simplex algorithm, TS greatly reduces the number of operations to maintain the constraint matrix by taking advantage of its special structure. The empirical study in [42] shows that the time complexity is super-cubic for signatures with size N. Other possible solutions mentioned in [42] include interior-point algorithms [20] and incapacitated minimum network flow [1] that have similar time complexities.

B. The EMD between Histograms

Histograms can be viewed as a special type of signatures in that each histogram bin corresponds to an element in a signature. In this view, the histogram values are treated as the weights w_j in a signature S, and the grid locations (indices of bins) are treated as positions m_j in S.

In the following we assume two dimensional histograms for illustrative simplicity. They are widely used for shape and image descriptors and derivations for higher dimensional cases are straightforward. Without loss of generality, we use the following assumptions and notations.

- Histograms have m rows and n columns and $N = m \times n$ bins.
- The index set for bins is defined as $\mathcal{I} = \{(i, j) : 1 \le i \le m, 1 \le j \le n\}$. We use (i, j) to denote a bin or a node corresponding to it.
- The index set for flows is defined as $\mathcal{J} = \{(i, j, k, l) : (i, j) \in \mathcal{I}, (k, l) \in \mathcal{I}\}$.
- $P = \{p_{ij} : (i, j) \in \mathcal{I}\}$ and $Q = \{q_{ij} : (i, j) \in \mathcal{I}\}$ are the two histograms to be compared.
- Histograms are normalized to a unit mass, i.e., $\sum_{i,j} p_{ij} = 1$, $\sum_{i,j} q_{ij} = 1$. As will be clear later, the normalization is not essential for the algorithm we will propose.

- The bin sizes in both dimensions are equal. Without loss of generality, each bin is assumed to be a unit square.

With these notations and assumptions, we obtain the following new definition of EMD between two histograms P and Q

$$EMD(P,Q) = \min_{F=\{f_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}\}} \sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} \tag{1}$$

$$\text{s.t.} \begin{cases} \sum_{(k,l)\in\mathcal{I}} f_{i,j;k,l} &= p_{ij} & \forall (i,j)\in\mathcal{I} \\ \sum_{(i,j)\in\mathcal{I}} f_{i,j;k,l} &= q_{kl} & \forall (k,l)\in\mathcal{I} \\ f_{i,j;k,l} &\geq 0 & \forall (i,j,k,l)\in\mathcal{J} \end{cases} \tag{2}$$

where F is a flow from P to Q and $f_{i,j;k,l}$ denotes a flow from bin (i,j) to (k,l). Note that we use the term "flow" to indicate both the set of flows in a graph and a single flow between two nodes, when there is no confusion. A flow F satisfying (2) is called *feasible*.

The ground distance $d_{i,j;k,l}$ is commonly defined by L_p distance

$$d_{i,j;k,l} = \|(i,j)^\top - (k,l)^\top\|_p = (|i-k|^p + |j-l|^p)^{1/p} \tag{3}$$

For example, the original EMD proposed by Rubner et al. [42] employed the L_1 (for texture) and L_2 (for color) ground distances.

IV. EMD-L_1

This section introduces EMD-L_1, a novel efficient formulation of EMD between histograms. We first show that, by using the L_1 (Manhattan) distance as the ground distance, EMD-L_1 drastically simplifies the original formulation. Then, we formally prove that EMD-L_1 is equivalent to the original EMD with L_1 ground distance. Note that we use the term EMD-L_1 to refer to the proposed formulation (and algorithms), which should be distinguished with the original EMD with L_1 ground distance.

A. Formulation of EMD-L_1

The robustness and efficiency of the L_1 norm often makes it preferable to the L_2 norm in computer vision and related areas, such as low-level vision learning [6], stereo analysis [19], [5], 1-norm support vector machine [51], etc. In addition, the L_1 and L_2 norms often perform similarly for image retrieval tasks [2]. Furthermore, the L_1 formulation had been also used in

EMD, such as in [50], [42], [4] etc.. Inspired by this evidence, we choose L_1 as EMD's ground distance. In the rest of the paper, unless indicated otherwise, the L_1 ground distance is implicitly assumed when dealing with EMD. With the L_1 ground distance, formula (3) becomes

$$d_{i,j;k,l} = |i-k| + |j-l|$$

Note that the ground distance takes only integer values now. For illustrative purpose, the flow index set $\mathcal{J}$ is divided into three disjoint subsets $\mathcal{J} = \mathcal{J}_0 \bigcup \mathcal{J}_1 \bigcup \mathcal{J}_2$, each of which corresponds to one of the following types of flows.

- $\mathcal{J}_0 = \{(i,j,i,j) : (i,j) \in \mathcal{I}\}$ is for flows between bins at the same location. We call this type of flows *self-flows* or *s-flows* for short.
- $\mathcal{J}_1 = \{(i,j,k,l) : (i,j,k,l) \in \mathcal{J}, d_{i,j;k,l} = 1\}$ is for flows between neighbor bins. We call this type of flows *n-flows*.
- $\mathcal{J}_2 = \{(i,j,k,l) : (i,j,k,l) \in \mathcal{J}, d_{i,j;k,l} > 1\}$ is for other flows that are called *f-flows* because of their *far* distances.

An important property of the L_1 ground distance is that every positive f-flow can be replaced by a sequence of n-flows. This is because the L_1 distance forms a shortest path system on the integer lattice. For example, given an f-flow $f_{i,j;k,l}$, $i \leq k, j \leq l$, the L_1 ground distance has the following decomposition

$$d_{i,j;k,l} = d_{i,j;i,l} + d_{i,l;k,l} = \sum_{j \leq x < l} d_{i,x;i,x+1} + \sum_{i \leq y < k} d_{y,l;y+1,l} \tag{4}$$

In other words, any L_1 shortest path from (i,j) to (k,l) can be decomposed into a sum of edges with ground distance one. It follows that, without changing the total weighted flow $\sum_{f \in F} fd$, the f-flow $f_{i,j;k,l}$ can be removed by increasing all n-flows along the path $[(i,j),(i,j+1),\ldots,(i,l),(i+1,l),\ldots,(k,l)]$ with $f_{i,j;k,l}$. This is illustrated in Fig. 3

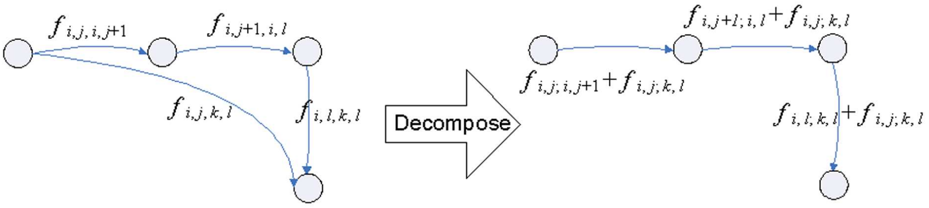


Fig. 3. Decompose an f-flow $f_{i,j;k,l}$, $k = i+1, l = j+2$. Only flows involved in decomposition are shown.

In addition to f-flows, s-flows can also be removed due to the zero ground distances associated with them while maintaining the total weighted flow. With these intuitions, we propose *EMD-L_1*: a new simplified formulation of EMD that only uses n-flows

$$EMD\text{-}L_1(P,Q) = \min_{G=\{g_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}_1\}} \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{5}$$

$$\text{s.t.} \left\{ \begin{array}{rcll} \sum_{k,l:(i,j,k,l)\in\mathcal{J}_1} (g_{i,j;k,l} - g_{k,l;i,j}) & = & b_{ij} & \forall (i,j) \in \mathcal{I} \\ g_{i,j;k,l} & \geq & 0 & \forall (i,j,k,l) \in \mathcal{J}_1 \end{array} \right. \tag{6}$$

where $b_{ij} = p_{ij} - q_{ij}$ is the difference between the two histograms at a bin (i,j). We call a flow G satisfying (6) a *feasible* flow, analogous to that in the original EMD. The intuition of constraint (6) is that, for a feasible flow G, the total flow that leaves any node (i,j) minus the total flow that enters (i,j) should be equal to b_{ij} (the difference between the two histogram bins).

EMD-L_1 is largely simplified compared to the original EMD in (1) and (2). The specific simplifications include

1) There are only $4N$ variables in (5), one order of magnitude less than that in (1). This is critical for speedup since the number of variables is a dominant factor in the time complexity of all LP algorithms. In addition, the memory efficiency gained by this is very favorable for histograms with a large number of bins.
2) The number of equality constraints is reduced by half. This is another important factor for deriving an efficient LP algorithm.
3) All the ground distances involved in the EMD-L_1 become ones. This is practically useful, because it removes all the distance computation and thus each flow g is equivalent to the corresponding weighted flow gd (d is the ground distance corresponds to the flow g). It also allows the use of integer operations to handle the coefficients.

EMD-L_1 can also be interpreted as a network flow model illustrated in Fig. 4. In the model, each bin (i,j) is treated as a node with weight b_{ij}, and eight flow edges (as shown in Fig. 4) between the node and its four neighbors. The total weight of the nodes is 0 ($\sum_{\mathcal{I}} b_{ij} = 0$). The task is to redistribute the weights via the flows to make all weights vanish. In this interpretation, EMD-L_1 is given by a solution with the minimum total flow.

The above simplifications and the network flow interpretation enable us to design a fast tree-based algorithm to solve EMD-L_1, which we present in Sec. V-C.

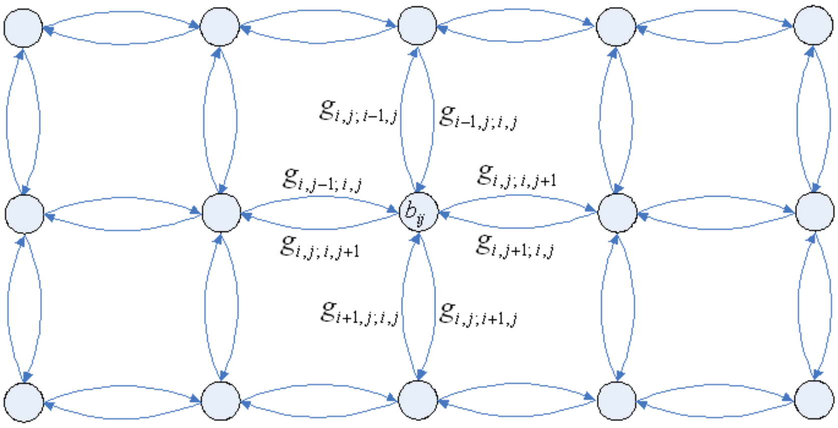


Fig. 4. The EMD-L_1 as a network flow problem for 3×5 histograms.

B. Equivalence between EMD-L_1 and Original EMD with L_1 Ground Distance

The equivalence here is in the sense of the weighted total flows. For example, a flow G for EMD-L_1 and a flow F in the original EMD is said to be equivalent if $\sum_{\mathcal{J}_1} g_{i,j;k,l} = \sum_{\mathcal{J}} d_{i,j;k,l} f_{i,j;k,l}$, i.e., they have same total weighted flow. The following proposition states the equivalence in which we are interested.

Proposition Given two histograms P and Q as defined above

$$EMD(P,Q) = EMD\text{-}L_1(P,Q) \ . \qquad (7)$$

We now introduce the intuition of the proof. The discussion in the last subsection suggests that, for any feasible flow F for the original EMD, an equivalent feasible flow G for EMD-L_1 (i.e. $\sum_{\mathcal{J}_1} g_{i,j;k,l} = \sum_{\mathcal{J}} d_{i,j;k,l} f_{i,j;k,l}$) can be created by eliminating all f-flows in F by using the decomposition and removing s-flows. This implies $EMD(P,Q) \geq EMD\text{-}L_1(P,Q)$. Now we need to verify the other direction. Given a flow G for EMD-L_1, find an equivalent F for the original EMD. The key issue is how to satisfy the constraints (2) in the original EMD. To do this, we introduce a “merge” procedure. The idea is to merge input and output flows at each bin so that either input or output flows disappear as a result. This is demonstrated in Fig. 5 on page 15. Notice that, for this proof, we only need an F to have a total weight no greater than that of G. This makes the proof with the merge procedure much simpler, allowing us to merge any pair of input and output flows.

Proof To prove (7), it suffices to prove

$$EMD(P,Q) \geq EMD\text{-}L_1(P,Q) \quad \text{and} \quad EMD(P,Q) \leq EMD\text{-}L_1(P,Q) \ .$$

Part I Proof of $EMD(P,Q) \geq EMD\text{-}L_1(P,Q)$.

It suffices to prove that for any feasible flow $F = \{f_{i,j;k,l} : (i,j,k,l) \in \mathcal{J}\}$ for the original EMD, there exists an equivalent feasible flow $G = \{g_{i,j;k,l} : (i,j,k,l) \in \mathcal{J}_1\}$ for EMD-L_1, i.e.

$$\sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} = \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{8}$$

This is because, if the above statement is true, we have

$$EMD(P,Q) = \min_F \sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} \ \geq \ \min_G \sum_{\mathcal{J}_1} g_{i,j;k,l} = EMD\text{-}L_1(P,Q)$$

where "$\geq$" is due to the above statement.

For any F satisfying (2), we create an auxiliary flow $F' = \{f'_{i,j;k,l:(i,j,k,l) \in \mathcal{J}}\}$. First, F' is initialized by F. F' has three properties which will be maintained during its evolution

$$\begin{cases} \sum_{\mathcal{J}} f'_{i,j;k,l} d_{i,j;k,l} & = & \sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} & \\ \sum_{k,l} (f'_{i,j;k,l} - f'_{k,l;i,j}) & = & b_{ij} & \forall (i,j) \in \mathcal{I} \\ f'_{i,j;k,l} & \geq & 0 & \forall (i,j,k,l) \in \mathcal{J} \end{cases} \tag{9}$$

Then, we evolve F' to make all f-flows vanish. For every positive f-flow $f'_{i,j;k,l}$ in F', we decompose it into a sequence of n-flows as illustrated in Fig. 3. In detail, assume $i \leq k, j \leq l$, the three modifications to F' are conducted as following in the given order

$$\begin{cases} f'_{i,x;i,x+1} & \leftarrow & f'_{i,x;i,x+1} + f'_{i,j;k,l} & \forall x, j \leq x < l \\ f'_{y,l;y+1,l} & \leftarrow & f'_{y,l;y+1,l} + f'_{i,j;k,l} & \forall y, i \leq y < k \\ f'_{i,j;k,l} & \leftarrow & 0 & \end{cases} \tag{10}$$

It is clear that (9) always holds before and after (10) (though it might be violated when (10) is only partially finished). A similar operation can be defined for other index inequality cases. After all the f-flows vanish, we build G from F'

$$g_{i,j;k,l} = f'_{i,j;k,l} \ , \ \forall (i,j,k,l) \in \mathcal{J}_1 \tag{11}$$

From (9), it follows that G satisfies (6) and (8) (due to the fact that $f'_{i,j;k,l} = 0, \ \forall (i,j,k,l) \in \mathcal{J}_0 \bigcup \mathcal{J}_2$). That is, G is a feasible flow for $EMD\text{-}L_1(P,Q)$ that is equivalent to F. Therefore, we have $EMD(P,Q) \geq EMD\text{-}L_1(P,Q)$.

Part II Proof of $EMD(P,Q){\le}EMD\text{-}L_1(P,Q)$.

Similar to Part I, it suffices to prove that, for any feasible flow $G=\{g_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}_1\}$ satisfying (6), there exists $F=\{f_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}\}$ satisfying (2), such that

$$\sum_{\mathcal{J}} f_{i,j;k,l}d_{i,j;k,l}\le\sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{12}$$

For any G satisfying (6), we create an auxiliary flow $G'=\{g'_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}\}$. G' is first initialized by G

$$g'_{i,j;k,l}=\begin{cases} g_{i,j;k,l} & \forall(i,j,k,l)\in\mathcal{J}_1\\ 0 & \forall(i,j,k,l)\in\mathcal{J}_0\bigcup\mathcal{J}_2\end{cases}$$

G' has three properties which will be maintained during its evolution

$$\left\{\begin{array}{rcll} \sum_{\mathcal{J}} g'_{i,j;k,l}d_{i,j;k,l} & \le & \sum_{\mathcal{J}_1} g_{i,j;k,l} & \\ \sum_{k,l\in\mathcal{I}}(g'_{i,j;k,l}-g'_{k,l;i,j}) & = & b_{ij} & \forall(i,j)\in\mathcal{I}\\ g'_{i,j;k,l} & \ge & 0 & \forall(i,j,k,l)\in\mathcal{J}\end{array}\right. \tag{13}$$

Note that, in the first equation of (13), “$\le$” is used instead of “$=$”.

Now we evolve G' targeting the equality constraints (2) in the original EMD. This is done by the following procedure.

```
*Procedure: Merge $G'$*
  FOR each grid node $(i,j)$
    WHILE exists flow $g'_{k,l;i,j}>0$ AND flow $g'_{i,j;k',l'}>0$ DO
```

$$\left\{\begin{array}{rcl} \delta & \leftarrow & \min\{g'_{i,j;k',l'},g'_{k,l;i,j}\}\\ g'_{k,l;k',l'} & \leftarrow & g'_{k,l;k',l'}+\delta\\ g'_{k,l;i,j} & \leftarrow & g'_{k,l;i,j}-\delta\\ g'_{i,j;k',l'} & \leftarrow & g'_{i,j;k',l'}-\delta\end{array}\right. \tag{14}$$

```
    END WHILE
  END FOR
```

Fig. 5 shows an example of merging. The four steps in (14) need to be applied in the order as given. Moreover, each run of (14) removes at least one non-zero flow, so the procedure is guaranteed to terminate. Note that the merged flow may not be unique. However this does not affect our proof because only total weighted flows are concerned.

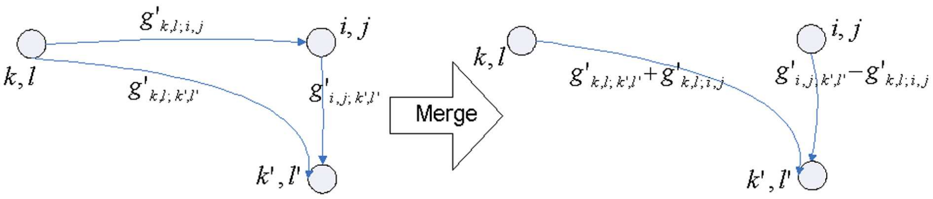


Fig. 5. Flow merging, where $b_{ij} > 0$, $g'_{i,j;k',l'} > g'_{k,l;i,j} > 0$.

Because of the triangle inequality $d_{k,l;k',l'} \leq d_{k,l;i,j} + d_{i,j;k',l'}$, the procedure (14) will not alter the first inequality in (13) since it only decrease its left hand side. The second equality in (13) also holds because (14) changes the input and output flows of a node always with the same amount (δ). The third condition in (13) obviously holds true.

An important observation due to (13) and the proposed merge procedure is

$$\begin{cases} g'_{i,j;k,l} & = & 0 & \forall(i,j,k,l) \in \mathcal{J} & \text{if } b_{ij} \leq 0 \\ g'_{k,l;i,j} & = & 0 & \forall(i,j,k,l) \in \mathcal{J} & \text{if } b_{ij} \geq 0 \end{cases} \tag{15}$$

Now we build F from G':

$$f_{i,j;k,l} = \begin{cases} \min\{p_{ij}, q_{kl}\} & \forall(i,j,k,l) \in \mathcal{J}_0 \\ g'_{i,j;k,l} & \forall(i,j,k,l) \in \mathcal{J}_1 \bigcup \mathcal{J}_2 \end{cases} \tag{16}$$

From (13), (15) and (16), we have that F satisfies (2) and (12). That is, F is a feasible for $EMD(P,Q)$ and $\sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} \leq \sum_{\mathcal{J}_1} g_{i,j;k,l}$. Therefore, $EMD(P,Q) \leq EMD\text{-}L_1(P,Q)$. ■

V. Algorithms for EMD-L_1

To compute EMD-L_1 between histograms is equivalent to solving the linear programming (LP) problem in (5) and (6). We designed a tree-based algorithm as an efficient discrete optimization solver, which extends the original simplex algorithm. The tree-based algorithm is significantly faster than the original simplex, and has a more intuitive interpretation as a network flow problem. As a reference, we will first briefly describe the standard simplex applied to EMD-L_1. After that, an extended transportation simplex algorithm for EMD-L_1 is designed based on the original transportation simplex [12] used in [42]. Finally, the tree-based algorithm is derived by further extending the fast simplex.

A. The Simplex Algorithm for EMD-L_1

The simplex algorithm is a popular solution to linear programming problems because of its average polynomial time complexity. In this subsection, we will first formulate EMD-L_1 as a standard linear program and then briefly describe its solution by the standard simplex algorithm. Detailed descriptions of the simplex algorithm can be found in any linear optimization book. We follow definitions and terminologies in [12].

First, write EMD-L_1 as a standard LP problem

$$\max \quad Z \tag{17}$$

$$\text{s.t.} \left\{ \begin{array}{rcll} Z + \sum_{\mathcal{J}_1} c_{i,j;k,l} g_{i,j;k,l} & = & 0 & \\ \sum_{k,l:(i,j,k,l)\in\mathcal{J}_1} \left(s_{ij} g_{i,j;k,l} - s_{ij} g_{k,l;i,j}\right) & = & |b_{ij}| & \forall (i,j) \in \mathcal{I} \\ g_{i,j;k,l} & \geq & 0 & \forall (i,j,k,l) \in \mathcal{J}_1 \end{array} \right. \tag{18}$$

where $s_{ij} = \text{sign}(b_{ij})$ is the sign of b_{ij}. The constraints can be written in matrix formulations,

$$\begin{pmatrix} 1 & \mathbf{c}^\top \\ \mathbf{0} & \mathbf{A} \end{pmatrix} \begin{pmatrix} Z \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix} \tag{19}$$

where

$$\mathbf{A} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & -s_{ij} & \dots & -s_{ij} & \dots & s_{ij} & s_{ij} & s_{ij} & s_{ij} & \dots & -s_{ij} & \dots & -s_{ij} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{20}$$

$$\begin{aligned} \mathbf{g} &= (\dots g_{i-1,j;i,j} \dots g_{i,j-1;i,j} \dots g_{i,j;i-1,j}, g_{i,j;i,j-1}, g_{i,j;i,j+1}, g_{i,j;i+1,j} \\ &\quad \dots g_{i,j+1;i,j} \dots g_{i+1,j;i,j} \dots)^\top \end{aligned} \tag{21}$$

$$\mathbf{b} = (\dots |b_{ij}| \dots)^\top \tag{22}$$

$$\mathbf{c} = (1 \dots 1)^\top \tag{23}$$

Notes: 1) In the above formulation, $Z = -\sum_{\mathcal{J}_1} c_{i,j;k,l} g_{i,j;k,l}$ is the negative of the original objective function, which makes the problem as a “maximization” problem. 2) The objective function is treated as the first row of the constraint matrix for convenience. 3) The coefficient **c** is not constant. Roughly speaking, it reflects the constraints between components of variable g and controls the iteration of the optimization. For more details, the readers are referred to [12, Chapter 4 and 5].

There are several observations on this LP formulation of the EMD-L_1:

- Each row of $\mathbf{A}$ corresponds to a node/bin in the histograms. Furthermore, only eight entries are non-zero for each row - four for input flows and the other four for outputs. A row corresponding to a bin (i,j) is shown in (20) and the corresponding flows are shown in (21) accordingly.
- Each column of $\mathbf{A}$ has only two non-zero entries that relate to the two ends of a flow. Specifically, for the column corresponding to $g_{i,j;k,l}$, only rows corresponding to the node (i,j) and (k,l) are non-zero.
- Although there are mn equality equations in (18), the actual number of constraints is $mn-1$ because $\sum b_{ij}=0$.

The simplex algorithm searches for the optimum solution among the space of *basic feasible (BF) solutions*. A BF solution is a solution of (18) such that only a fixed number of variables can be non-zero. These variables are called *basic variable* (BV) flows in our formulation and we use $\mathcal{B}$ to denote the set of BV flows. The number of BV flows corresponds to the number of constraints in the LP problem ($mn-1$ for EMD-L_1). The simplex algorithm employs an iterative optimization approach: given an initial BF solution, it iteratively finds a better BF solution (and replaces the old one) until the optimum is reached. Intuitively, each BF solution lies at an intersection of the constraint boundaries (hence the number of BF solutions is finite). The simplex iteration is guaranteed to converge to the global optimum because of the convexity of the constraints [37, Theorem 2.9, p53-54].

The coefficient $\mathbf{c}$ is very important in three ways. First, the algorithm reaches optimum *iff* all elements of $\mathbf{c}$ are non-negative. This is used as the termination criterion for the iteration in the simplex algorithm. Second, $c_{i,j;k,l}$ vanishes for every BV flow $g_{i,j;k,l}$ in a BF solution $\mathbf{g}$. Third, the most negative element in $\mathbf{c}$ is used to determine how to improve the current BF solution.

The key to the algorithm is to find a better BF solution than the current one. The new BF solution has only one different BV flow than the current one. In other words, in each iteration, one flow leaves $\mathcal{B}$ and one flow enters $\mathcal{B}$ (from outside $\mathcal{B}$). The flow leaving $\mathcal{B}$ is called *leaving BV* and denoted as $g_{i_1,j_1;k_1,l_1}$. Accordingly, the flow entering $\mathcal{B}$ is called *entering BV* and denoted as $g_{i_0,j_0;k_0,l_0}$. During each iteration, the simplex algorithm first finds $g_{i_0,j_0;k_0,l_0}$ outside $\mathcal{B}$ by some greedy criteria. Then $g_{i_1,j_1;k_1,l_1}$ that achieves the maximum improvement of $g_{i_0,j_0;k_0,l_0}$ is found inside $\mathcal{B}$. Table I outlines the simplex algorithm applied to EMD-L_1. Details of the terminologies, as well as the simplex algorithm, can be found in [12, Chapter 4].

TABLE I

SIMPLEX ALGORITHM FOR EMD-L_1

```
Step 1  /* Initialization */
        Initialize matrix $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$
        Find the initial BF solution $\mathbf{g}$
        Update $\mathbf{c}$ and $\mathbf{A}$ according to $\mathbf{g}$
Step 2  /* Iteration */
        WHILE (1)
            /*Optimality test*/
            IF ($c_{i,j;k,l} \geq 0, \ \forall (i,j,k,l) \in \mathcal{J}_1$)
                $\mathbf{g}$ is optimal, goto Step 3
            END IF

            /*Find a new improved BF solution*/
            Find entering BV flow $g_{i_0,j_0;k_0,l_0}$ by the formula
                $(i_0, j_0, k_0, l_0) = \text{argmin}_{(i,j,k,l) \in \mathcal{J}_1} c_{i,j;k,l}$
            Find the leaving BV flow $g_{i_1,j_1,k_1,l_1}$ by the Minimum Ratio
                Test [12]. The intuition is to achieve the maximum
                improvement on $g_{i_0,j_0;k_0,l_0}$
            Use the elementary row operations (Gaussian eliminations)
                to update system (19), including $\mathbf{A}$, $\mathbf{c}$ and a new
                BF solution $\mathbf{g}$.
        END WHILE
Step 3  Compute the total flow by formula (5) as the EMD distance.
```

B. Extended Transportation Simplex for EMD-L_1

The original EMD is solved by the transportation simplex (TS) algorithm [12, Chapter 8] by taking advantage of the special structure of the original EMD formulation. As mentioned in the last subsection, EMD-L_1 has a sparse structure of the constraint matrix $\mathbf{A}$, which is similar to the original EMD. To exploit this similarity, we designed an *extended transportation simplex* (ETS) for EMD-L_1.

The basic idea of ETS is to intelligently update the BF solution $\mathbf{g}$, $\mathbf{c}$ and $\mathbf{A}$ during the simplex iteration. Notice that in the iteration, only row operations are applied on the constraint equation (19). Hence, instead of storing the whole matrix $\mathbf{A}$, we only need to keep track of multiples of each row when updating $\mathbf{c}$. For this reason, a new vector $\mathbf{v} = (\ldots, v_{ij}, \ldots)^\top$ is used, where

v_{ij} represents current multiples of the row corresponding to the node (i, j). As a consequence, coefficients $\mathbf{c}$ is updated by

$$c_{i,j;k,l} \quad = \quad 1 - s_{ij}v_{ij} + s_{kl}v_{kl} \qquad \forall(i, j, k, l) \in \mathcal{J}_1 \tag{24}$$

Notice that v_{ij} is always coupled with s_{ij}, so we merge them to form a *signed multiple vector* $\mathbf{u} = (\ldots, u_{ij} = s_{ij}v_{ij}, \ldots)^\top$. Then equation (24) is simplified to

$$c_{i,j;k,l} \quad = \quad 1 - u_{i,j} + u_{k,l} \quad \forall(i, j, k, l) \in \mathcal{J}_1 \tag{25}$$

Now the problem reduces to solving equation (25) about $\mathbf{u}$ and $\mathbf{c}$, and updating $\mathbf{g}$ accordingly. Notice that $c_{i,j;k,l}$ vanishes for every BV flow $g_{i,j;k,l}$, therefore

$$c_{i,j;k,l} \quad = \quad 1 - u_{i,j} + u_{k,l} = 0 \qquad \forall \text{ BV flow } g_{i,j;k,l} \tag{26}$$

Since there are $mn - 1$ BV flows and mn unknown u_{ij}, u_{ij} can be solved very efficiently using the special structure of (26). First, pick one u_{ij} (e.g, u_{11}) and set it to 0. Then, starting from it, we keep applying (26) until all other u_{ij} are solved. Once $\mathbf{u}$ is determined, $\mathbf{c}$ can be solved using (25).

Finding a better BF solution from the current BF solution $\mathbf{g}$ is not straightforward. First, the entering BV $g_{i_0,j_0;k_0,l_0}$ is found using the same procedure as in the original simplex algorithm, i.e., $(i_0, j_0; k_0, l_0)$ satisfies that $c_{i_0,j_0;k_0,l_0} = \min_{(i,j,k,l)\in\mathcal{J}_1} c_{i,j;k,l}$. Then, to find the leaving BV, we search for a loop in the BV flows starting from $g_{i_0,j_0;k_0,l_0}$. The loop is a sequence of BV flows $g_{r_0,c_0;r_1,c_1}g_{r_1,c_1;r_2,c_2}\cdots g_{r_L,c_L;r_0,c_0}$ where $r_0 = i_0, c_0 = j_0, r_1 = k_0, c_1 = l_0$. The existence and uniqueness of this loop is guaranteed. This loop contains all the BV flows to be updated in order to include $g_{i_0,j_0;k_0,l_0}$ into the new BF solution. Finally, the leaving BV flow $g_{i_1,j_1;k_1,l_1}$ is chosen from the loop, which has the minimum flow value and a reverse direction to $g_{i_0,j_0;k_0,l_0}$. For example, in Fig. 6 (b), the entering BV creates a loop when combined with current non-zero flows (the second and third columns from left). Among all the edges in this loop that have reversed directions to the entering BV, the one on the top is chosen as the leaving BV because it has the minimum flow value (0.2).

Table II lists the ETS algorithm. For better understanding, we recommend readers to refer to the original transportation simplex described in [12, Chapter 8].

TABLE II

EXTENDED TRANSPORTATION SIMPLEX (ETS) ALGORITHM FOR EMD-L_1

```
Step 1  /* Initialization */
        Initialize $\mathbf{b}$
        Find the initial BF solution $\mathbf{g}$
        Update $\mathbf{u}$ and $\mathbf{c}$ according to $\mathbf{g}$
Step 2  /* Iteration */
        WHILE (1)
            /*Optimality test*/
            IF ($c_{i,j;k,l} \geq 0,\ \forall (i,j,k,l) \in \mathcal{J}_1$)
                $\mathbf{g}$ is optimal, goto Step 3
            END IF

            /*Find a new improved BF solution*/
            Find entering BV flow $g_{i_0,j_0;k_0,l_0}$ by the formula
                $(i_0, j_0, k_0, l_0) = \text{argmin}_{(i,j,k,l)\in\mathcal{J}_1} c_{i,j;k,l}$
            Find a loop starting from the entering BV $(i_0, j_0, k_0, l_0)$
            Find the leaving BV $g_{i_1,j_1;k_1,l_1}$ as the one with the minimum
                flow value and a reverse direction in the loop as $g_{i_0,j_0;k_0,l_0}$.
            Update $\mathbf{g}$ along the loop, remove $g_{i_1,j_1;k_1,l_1}$ from $\mathcal{B}$
                and add $g_{i_0,j_0;k_0,l_0}$ into $\mathcal{B}$.
            Update $\mathbf{c}$ using formula (25).
        END WHILE
Step 3  Compute the total flow by formula (5) as the EMD distance.
```

C. Tree-EMD

Now consider the structure of a BF solution from the viewpoint of the network flow interpretation of EMD-L_1, which was mentioned in Sec. IV-A and in Fig. 4. There are two useful facts of ETS as listed below.

1) There are mn nodes in the network and only $mn - 1$ non-zero flows in a BF solution.
2) An optimal BF solution contains no cycles.

These facts suggest that a BF solution forms a *spanning tree* in the network graph. In the following, we call such a tree a *basic feasible tree* (BFT). Fig. 6 (a) shows an example of a BFT. As a result, an efficient solution of EMD-L_1 can be designed to find a BF tree with minimum total tree weight (flows). Note that BF trees are undirected trees though flows do have directions (as shown in Fig. 6). In other words, when talking about cycles in this subsection, we mean undirected cycles.

With this tree-based formulation, the iteration in ETS has a new interpretation. The entering BV $g_{i_0;j_0;k_0,l_0}$ is an edge to be added to the tree to reduce the total flow. A loop is formed after adding $g_{i_0;j_0;k_0,l_0}$. The leaving BV $g_{i_1,j_1;k_1,l_1}$ is the minimum edge in the loop that has a direction reversed from $g_{i_0;j_0;k_0,l_0}$.

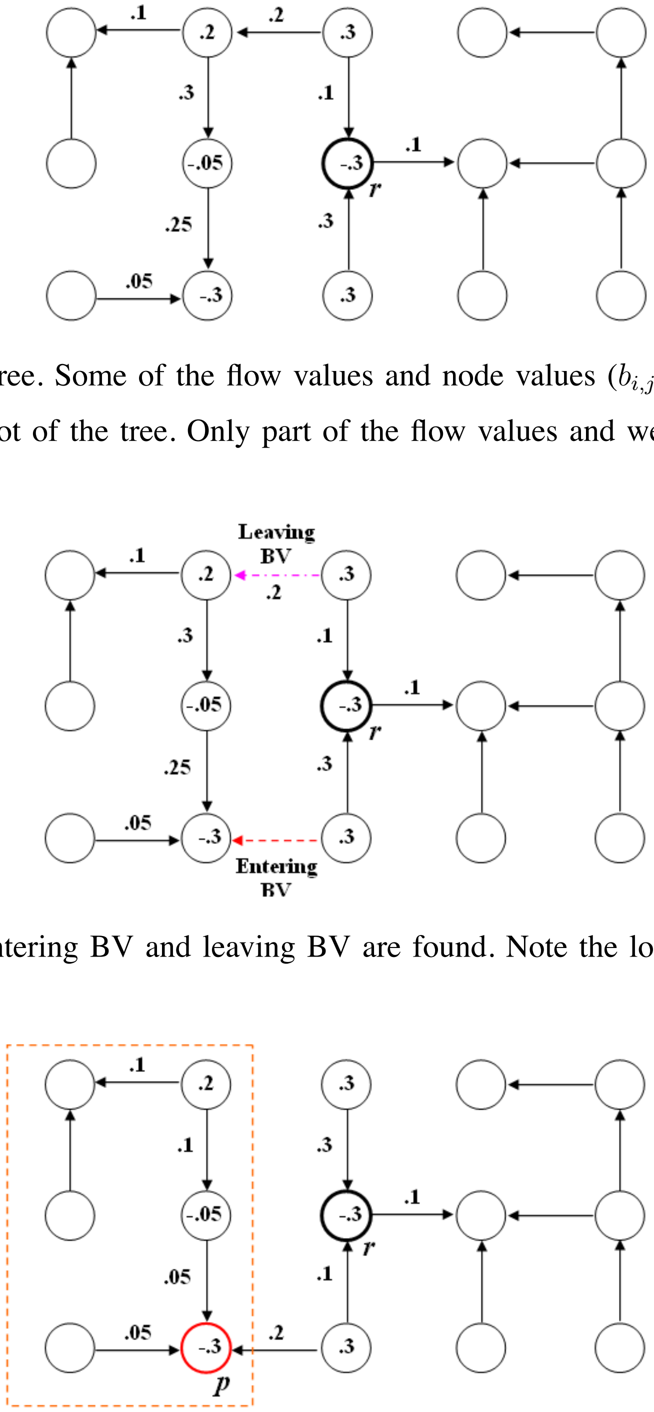
A tree-based algorithm, *Tree-EMD*, can be naturally extended from ETS. First, an initial BFT is built. Then the BFT is iteratively replaced by a better BFT until the optimum is reached. Compared to ETS, Tree-EMD is more efficient due to the following reasons.

- Finding the loop from the $g_{i_0,j_0;k_0,l_0}$ in transportation simplex requires graph searching [12]. This can be very slow (exponential worst complexity), especially for large histograms. A tree-based algorithm can solve this problem efficiently, since the cycle containing $g_{i_0;j_0;k_0,l_0}$ can be easily identified by tracing from node (i_0, j_0) and (k_0, l_0) until finding their latest common ancestor. This is very efficient because it avoids the brute force search used in the ETS algorithm [12, p327-328].
- With a tree structure, there is no need to update the whole $\mathbf{u}$. Only u_{ij} in a subtree needs to be updated. This is true because u_{ij} only depends on their parents and we can always set u_{ij} to 0 for the root. In addition, we also avoid locating unsolved u_{ij} as required in the transportation simplex algorithm [12, p328].

An example tree updating in one iteration is illustrated in Fig. 6. Fig. 6 (b) shows the entering BV and leaving BV found from the tree in (a). Fig. 6 (c) shows the new improved tree after removing the leaving BV and adding the entering BV. In addition, an edge p is shown to indicate the root of the subtree where $\mathbf{u}$ need to be updated.

The Tree-EMD algorithm is presented in Table III. Several issues are discussed below.

1) *The root of a BFT*: The root r is heuristically set to be the center of the graph. This is to make the tree as balanced as possible. Once r is fixed, the u value at r is fixed to 0.
2) *Build the initial BFT*: For this task, we designed a greedy algorithm that is listed in Table IV. The nodes are considered sequentially, in a left-to-right and bottom-to-top order, i.e., starting from bottom-left node. When processing node q, all the flows connecting its lower and left neighbors are fixed. As a result, only one BV flow needs to be chosen between q and either its upper or right neighbor such that the flow makes the weight at q vanish.



(a) A BF tree. Some of the flow values and node values ($b_{i,j}$) are listed. r denotes the root of the tree. Only part of the flow values and weights are shown.

(b) The entering BV and leaving BV are found. Note the loop formed.

(c) The improved BF tree. p is the root of the subtree where $\mathbf{u}$ need to be updated. The subtree is indicated in the dashed bounding box.

Fig. 6. Tree updating in Tree-EMD algorithm.

TABLE III

TREE-EMD

```
Step 1  /* Initialization */
        Initialize $\mathbf{b}$
        Build the initial BFT $\mathbf{g}$ rooted at $r$ by a greedy initial solution (Table IV)
        $r \leftarrow$ the center of the graph      /* $r$ is the root of the tree */
        $p^* \leftarrow r$      /* $p^*$ is the root of the subtree to be updated */

Step 2  /* Iteration */
        WHILE(1)
            /*Recursively update $\mathbf{u}$ in the subtree rooted at $p^*$)*/
            FOR any child $q$ of $p^*$
                Update $u_{ij}$ at node $q$ according to (26)
                Recursively update $q$'s children
            END IF

            /*Optimality test*/
            IF ($c_{i,j;k,l} \geq 0, \ \forall(i,j,k,l) \in \mathcal{J}_1$)
                $\mathbf{g}$ is optimal, goto Step 3
            END IF

            /*Find a new improved BF solution*/
            Find entering BV flow $g_{i_0,j_0;k_0,l_0}$ by the formula
                $(i_0, j_0, k_0, l_0) = \text{argmin}_{(i,j,k,l) \in \mathcal{J}_1} c_{i,j;k,l}$
            Find loop by tracing from node $(i_0, j_0)$ and $(k_0, l_0)$ to find their latest ancestor.
            Find the leaving BV $g_{i_1,j_1;k_1,l_1}$ as the one with the minimum flow value and
                a reverse direction in the loop as $g_{i_0,j_0;k_0,l_0}$.
            Update $\mathbf{g}$ along the loop
            Maintain the tree, include removing $g_{i_1,j_1;k_1,l_1}$ from it and adding $g_{i_0,j_0;k_0,l_0}$ into it,
                together with related parent-child linkages.
            Update $\mathbf{c}$ using formula (25).
            Set $p^*$ as the root of subtree to update $\mathbf{u}$
        END WHILE

Step 3  Compute the total flow by formula (5) as the EMD distance.
```

The choice is based on which direction is more effective in making the rest of the nodes "even" (i.e., with smaller total absolute weights, see Table IV for details). Note that this

TABLE IV

Greedy-Solution for initializing BFT. Note. 1) The summation $\Sigma_{r+1 \le i \le m, 1 \le j \le n} b'_{ij}$ and $\Sigma_{1 \le i \le m, c+1 \le j \le n} b'_{ij}$ can be computed dynamically for efficiency. 2) A BV flow can have zero value. 3) The topmost row and rightmost column may be treated separately, here we prefer the concise description for clearness.

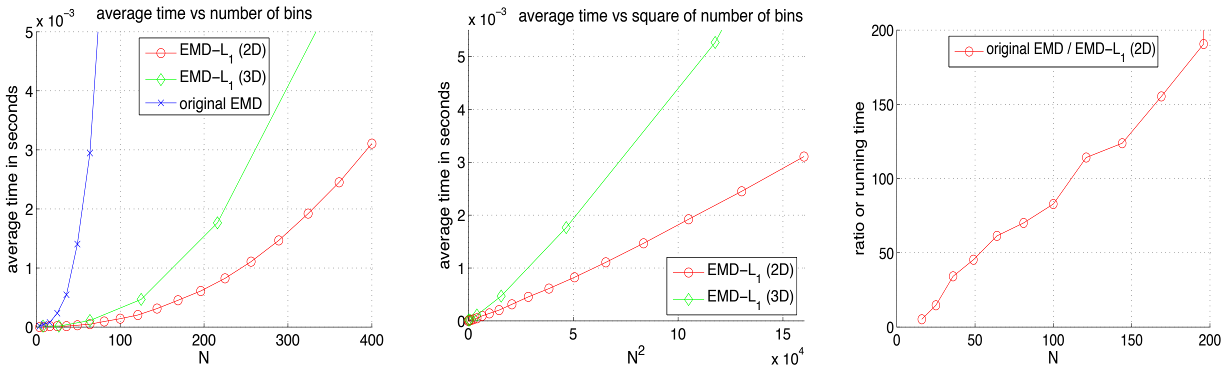
```
Step 1  /* Initialize all the flows */
        $g_{i,j;k,l} \leftarrow 0, \quad \forall(i,j,k,l) \in \mathcal{J}_1$
        $b'_{i,j} \leftarrow bij, \quad \forall(i,j) \in \mathcal{I}$     /* residual weights */

Step 2  /* Greedily find BV flows */
        FOR $c = 1 : n$
            FOR $r = 1 : m$
                IF $r \neq m$ AND $c \neq n$
                    IF $|b'_{r,c} + \Sigma_{r+1 \le i \le m, 1 \le j \le n} b'_{ij}| < |b'_{r,c} + \Sigma_{1 \le i \le m, c+1 \le j \le n} b'_{ij}|$
                        /* Flow to or from up */
                        IF $b'_{r,c} > 0$    $g_{r,c;r+1,c} \leftarrow b'_{r,c}$
                        ELSE               $g_{r+1,c;r,c} \leftarrow b'_{r,c}$
                        END IF
                        $b'_{r+1,c} \leftarrow b'_{r+1,c} + b'_{r,c}$
                    ELSE
                        /* Flow to or from right */
                        IF $b'_{r,c} > 0$    $g_{r,c;r,c+1} \leftarrow b'_{r,c}$
                        ELSE               $g_{r,c+1;r,c} \leftarrow b'_{r,c}$
                        END IF
                        $b'_{r,c+1} \leftarrow b'_{r,c+1} + b'_{r,c}$
                    END IF
                END IF
            END FOR
        END FOR
```

approach can be easily extended for dimensions higher than two. A similar idea is also used for initialization of the transportation simplex, i.e. the *northwest corner rule* and the *Russel's initialization* [12, p320-324].

D. Empirical Study of Time Complexity

The simplex algorithm is known to have good empirical time complexity but poor worst case time complexity. Therefore, to evaluate the time complexity of the proposed algorithm, we conduct an empirical study similar to that in [42]. First, two sets of 2D random histograms are generated for sizes: $n \times n$, $2 \leq n \leq 20$. For each n, 1000 random histograms are generated for each set (i.e. 2000 for all). Then, the two sets are paired and the average time to compute EMD for each size n is recorded. We compare EMD-L_1 (with Tree-EMD) and the original EMD (with the TS algorithm[2]). In addition, EMD-L_1 is tested for 3D histograms with similar settings, except using $2 \leq n \leq 8$. In summary, three algorithms are compared: EMD-L1 for 2D, EMD-L1 for 3D, and the original EMD. The results are shown in Fig. 7. From (a) it is clear that EMD-L_1 is much faster than the original one. Fig. 7 (b) shows that EMD-L_1 has a complexity of $O(N^2)$, where N is the number of bins (n^2 for 2D and n^3 for 3D). Furthermore, in our image feature matching experiments (Sec. VI-B), EMD-L_1 shows similar running time as the quadratic form distance (see Table VII), which has a quadratic time complexity.



(a) (b) (c)

Fig. 7. Empirical time complexity study of EMD-L_1 (Tree-EMD). (a) In comparison to the original EMD (TS Algorithm). (b) Average running time vs. square of histogram sizes. (c) The ratio of the running time, i.e. $\frac{\text{running time of the original EMD}}{\text{running time of EMD-}L_1\text{(2D)}}$.

In addition to the above experiment, we also compared Tree-EMD and ETS in a pilot experiments for 2D histograms with 80 bins. We observed that Tree-EMD is roughly six times faster than the ETS algorithm.

By far EMD-L_1 has been shown to be more efficient than the original EMD. However, for sparse histograms, especially in high-dimensional spaces, the original EMD might have an advantage as it uses signatures that can compactly represent the sparse spaces with a relatively low number of features (bins).

[2] With Rubner's code, http://ai.stanford.edu/∼rubner/emd/default.htm

VI. Experiments

In this section, the EMD-L_1 is evaluated for various histogram-based local descriptors in two vision tasks. The first task is shape matching, where EMD-L_1 is used to compare shape context [3] and inner-distance shape context [26]. The second task is image feature (interest point) matching, comparing a number of distance metrics with SIFT [29], shape context [3] and spin image [18], [22]. These experiments show that EMD-L_1 is robust to the quantization problems [42] induced by geometrical deformation, illumination change and heavy noise.

A. Shape Matching with Shape Contexts

The EMD-L_1 is tested for shape matching with shape context (SC) [3] and the inner-distance shape context (IDSC) [26]. Given a shape and its boundary landmark points, SC attaches with each point, say p, with a histogram that measures the spatial distribution of all other points according to p's local coordinate system. IDSC is an extension of SC by using the shortest path distance inside the shape for distance bins instead of Euclidean distance. In [26], SC and IDSC are used for contour comparison with a dynamic programming (DP) scheme. We use the same experimental framework, except for replacing χ^2 distance with the EMD-L_1 for measuring dissimilarity between (inner-distance) shape contexts. We choose two shape data sets that have been previously used by other studies for comparison.

One useful property of the EMD is that it has a lower bound that can be efficiently computed with linear complexity [42]. This is used in our dynamic programming scheme to avoid computing the EMDs larger than twice the occlusion penalties. Also note that we did not test the EMD with L_2 ground distances for shape matching due to its high time requirement.

1) Articulated Database: The articulated shape data set [26] contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees (see Figure 8). This data set is designed for testing articulation, which is a special and important case of deformation. As shown in [26], the original shape context with χ^2 distance does not work well for these shapes. The reason is that the articulation incurs a large deformation in the histogram.

The experimental setup, also used in [26], is as follows. Two hundred points are sampled along the outer contours of every shape; five log-distance bins and twelve orientation bins are used for constructing SC and IDSC. For both shape representations, the same dynamic programming matchings are used to compute distances between pairs of shapes.

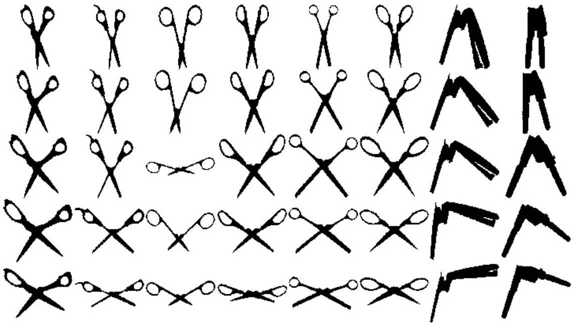


Fig. 8. Articulated shape database. This dataset contains 40 images from 8 objects. Each column contains five images of the same object with different articulation.

To evaluate the recognition result, for each image, the four most similar matches are computed from the dataset. Table V shows the retrieval results. The retrieval results are summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. It demonstrates that EMD-L_1 works better than the originally used χ^2 distance. Note that the improvement for IDSC is less impressive than for SC. This is because IDSC is articulation invariant by itself and there is not much room for improvement by using a different metric.

TABLE V

RETRIEVAL RESULT ON THE ARTICULATE DATASET.

SC	Top 1	Top 2	Top 3	Top 4	IDSC	Top 1	Top 2	Top 3	Top 4
χ^2 [26]	20/40	10/40	11/40	5/40	χ^2 [26]	40/40	34/40	35/40	27/40
EMD-L_1	**36/40**	**19/40**	**10/40**	**8/40**	**EMD-L_1**	**39/40**	**39/40**	**34/40**	**32/40**

2) MPEG7 CE-Shape-1: The MPEG7 CE-Shape-1 database [23] has been widely used to test various shape matching algorithms. The data set contains 1400 silhouette images from 70 classes. Each class has 20 different shapes (see Figure 9 for some typical images). The performance of different solutions is measured by the Bullseye test; every image in the database is matched with all other images and the top 40 most similar candidates are counted. At most 20 of the 40 candidates are correct hits. The Bullseye score is the ratio of the number of correct hits of all images to the best possible number of hits (which is 20×1400).

In this experiment, again we use the same setup as in [26] and replace the χ^2 metric with EMD-L_1. The bullseye score is listed in Table VI along with the previously reported results.

From the table it is clear that EMD-L_1 improves the performance of the IDSC. Moreover, the results of IDSC with EMD-L_1 outperforms the best reported score in the literature using the same data set, demonstrating our method's effectiveness.

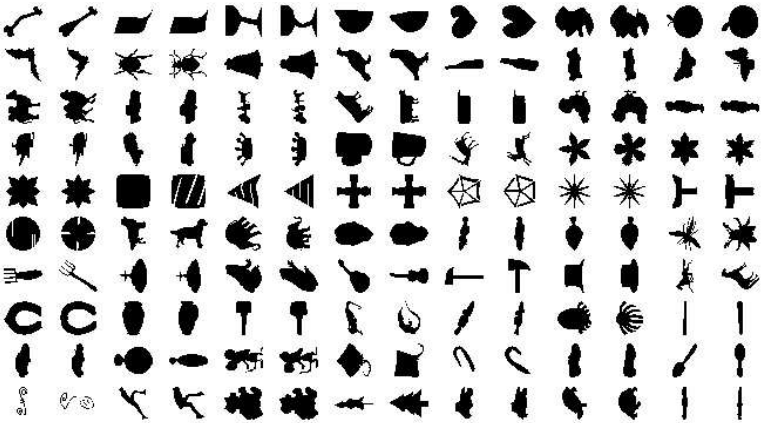


Fig. 9. Typical shape images from the MPEG7 CE-Shape-1, two images per class.

TABLE VI

RETRIEVAL RATE (BULLSEYE) OF DIFFERENT METHODS FOR THE MPEG7 CE-SHAPE-1.

Algorithm	CSS [33]	Visual Parts [23]	SC+TPS (χ^2) [3]	Curve Edit [43]	Distance Set [9]
Score	75.44%	76.45%	76.51%	78.17%	78.38%
Algorithm	MCSS [17]	Gen. Model [48]	IDSC (χ^2) [26]	**IDSC (EMD-L_1)**	
Score	78.8%	80.03%	85.40%	86.56%	

B. Image Feature Matching

This subsection describes our experiments for interest point matching with several state-of-the-art image descriptors. The experiment was conducted on two image data sets. The first data set contains ten image pairs with *synthetic* deformation[3], noise and illumination change. These images are shown in Fig. 10. The second set contains six image pairs with *real* deformation and lighting changes, as shown in Fig. 11. The experimental setting and results are described below.

[3] Original images are chosen from the the Berkeley Segmentation Dataset.

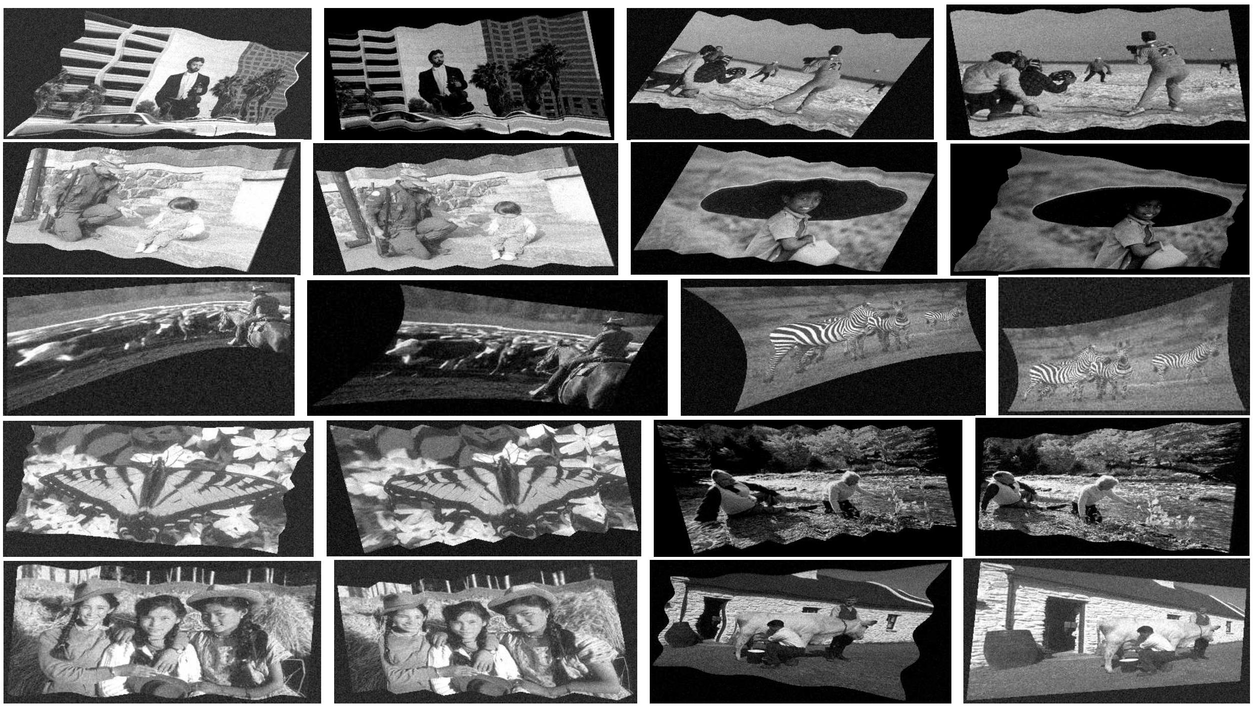


Fig. 10. Ten image pairs containing synthetic deformation, lighting change and noise.

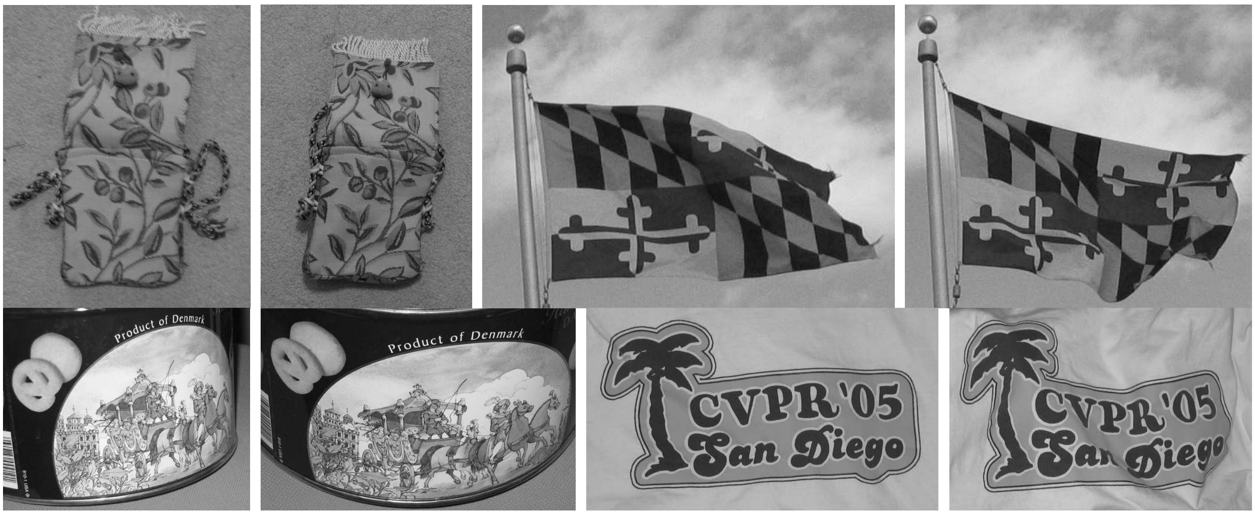


Fig. 11. Four of the six image pairs containing real deformation and lighting change.

Dissimilarity measures. We tested EMD-L_1 along with several popular bin-to-bin distances, as well as some other cross-bin distances. The bin-to-bin distances include χ^2 distance, symmetric

Kullback-Leibler divergence (KL), symmetric Jensen-Shannon (JS) divergence [25] and the L_2 distance. The cross-bin distances include EMD (with the L_2 ground distance) and quadratic form (QF). For EMD, we use the code provided by Rubner et al. [42]. The quadratic form distance is implemented according to [42]. We use the Tree-EMD algorithm for EMD-L_1 .

Interest point. We use Harris corners [11] for locating interest points. The reason for this choice is that other interest point detectors tend to fail for our data due to the large deformation, noise and lighting change. This choice also allows us to focus more on comparing descriptors than the interest point detection. For the synthetic data set, we computed 200 points per image pair with the largest cornerness responses. To compute the descriptors, a circular support region around each interest point is used. The region diameter is 41 pixels, which is similar to the setting used in [31].

Descriptors. We tested the above distances on three different histogram-based local descriptors. The first one is SIFT proposed by [29]. It is a weighted three dimensional histogram with four bins for each spatial dimensions and eight bins for gradient orientation. The second one is the shape context [3]. The shape context for images is extracted as a two dimensional histogram counting the local edge distribution in a manner similar to [31]. In our experiment, we use eight bins for distance and sixteen bins for orientation. The third one is the spin image [22], [18] which measures the joint spatial and intensity distribution of pixels around interest points. We use eight distance bins and sixteen intensity bins.

Evaluation criterion. For each pair of images with their interest points, we first find the ground-truth correspondence. This is done automatically for the synthetic image pairs and manually for the real image pairs. For efficiency, we removed the points in Image 1 without any correct corresponding matches in Image 2. This also makes the maximum detection rate one. After that, every interest point in Image 1 is compared with all interest points in Image 2 by comparing the features extracted on them. The detection rate among the top N matches is used to study the performance. The detection rate r is defined similarly to [27]

$$r = \frac{\text{\# correct matches}}{\text{\# possible matches}} = \frac{\text{\# correct matches}}{\text{\# valid points in Image 1}}$$

Experimental results. For evaluation, a performance curve for each distance measure is plotted showing the detection rates versus N, which is the number of the most similar matches allowed. The curves on the synthetic and real image pairs are shown in Fig. 12. The average time for comparing a real image pair is recorded and listed in Table VII. From the results it is clear that cross-bin distances, especially EMDs, outperform bin-to-bin distances. Among the cross-bin distances, both the proposed EMD-L_1 and EMD with L_2 demonstrate excellent accuracy, while the former runs much faster. It also shows that, on average, EMD-L_1 works better than the other cross-bin quadratic form distance that has similar time complexity.

TABLE VII

AVERAGE TIME (IN SECONDS) FOR INTEREST POINT MATCHING BETWEEN A REAL IMAGE PAIR. SC IS SHORT FOR SHAPE CONTEXT AND SI FOR SPIN IMAGE.

Approach	SIFT [29]	SC [3]	SI [22], [18]
χ^2	0.05	0.04	0.03
L_2	0.008	0.008	0.008
KL	0.16	0.21	0.19
JS	0.32	0.27	0.28
QF	3.56	3.47	3.52
EMD-L_1	5.76	3.54	3.58
EMD(L_2)	568.15	397.88	446.02

VII. CONCLUSION AND DISCUSSIONS

We propose EMD-L_1, a novel solution for computing Earth Mover's Distance (EMD) between histograms with L_1 ground distance. It reformulates the EMD into a drastically simplified version by using the special structure of the L_1 metric on histograms. The highlight is that the number of unknown variables in the optimization problem is reduced from $O(N^2)$ to $O(N)$, where N is the number of bins. We proved that EMD-L_1 is equivalent to the EMD with L_1 ground distance. Furthermore, an efficient tree-based algorithm is designed to solve the EMD-L_1. An empirical study shows that EMD-L_1 drastically improves the typical time complexity to $O(N^2)$ from $O(N^3 logN)$ of the original EMD. This speedup allows the EMD to be applied to compare 2D/3D histogram-based local features for the first time. Experiments on both shape descriptors (shape context [3]) and image features (SIFT [29], shape context [3] and spin images [18], [22]) show the superiority of EMD-L_1 for solving the two matching tasks with large deformation, noise and lighting change. It is also noteworthy that EMD-L_1 applied to IDSC outperformed the best reported system in solving the shape matching task with MPEG7 CE-Shape-1 data set.

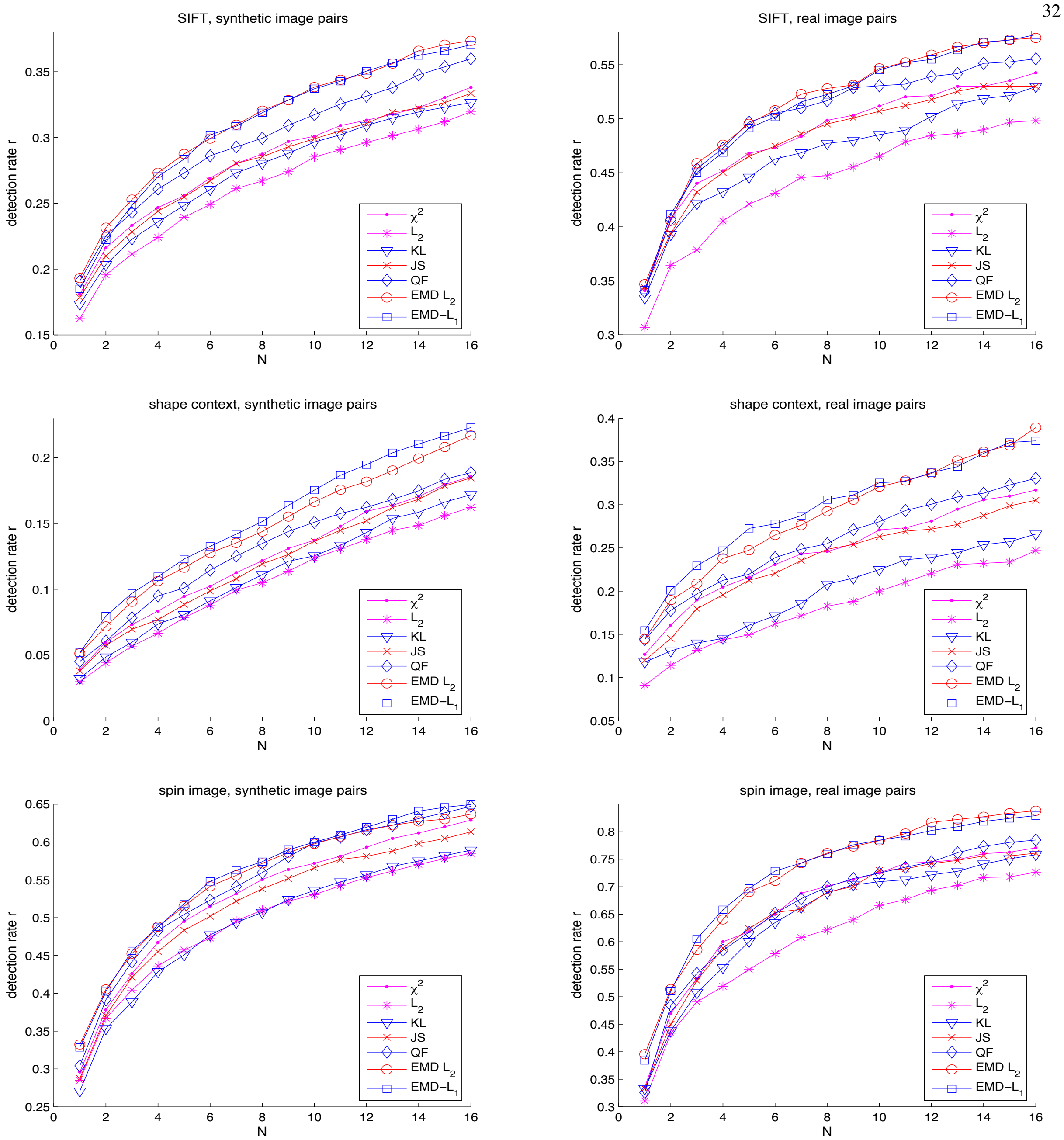


Fig. 12. Performance curves for interest point matching experiments. The left column is for synthetic image pairs and the right for real image pairs. The first row is for experiments with SIFT [29], the second for shape context [3], and the third for spin image [22]. In each figure, the y axis denotes the detection rate and x axis denotes the number of most similar matches allowed.

There are several issues that may lead to interesting future work. First, the normalization assumption of the histograms is not essential for EMD-L_1. This is because neither the formulation of EMD-L_1 nor its proposed algorithms is limited to the normalization case. Therefore, EMD-L_1

is also applicable to histograms with unequal total values. As a result, similar to the original EMD, EMD-L_1 has the potential ability to deal with occlusion, which is an important problem in local descriptors. Second, the Tree-EMD algorithm can also be generalized to solve the original EMD problem (i.e. beyond histograms) for speedup. This is because the tree structure used in Tree-EMD is also true for the transportation simplex used in the original EMD. In addition, as indicated in [42], EMD can also be modelled as a network flow problem. This raises interest in the underlying relationship between the tree-based algorithm and network flow algorithms. It may be a key to find more efficient solutions to both the original EMD and EMD-L_1.

Acknowledgments

We would like to thank David W. Jacobs for stimulating discussions and invaluable comments. We also thank Kevin S. Zhou, Doru-Cristian Balcan and Leo Grady for their useful comments. We thank Yossi Rubner for the transportation simplex code. Many thanks to the anonymous referees for their helpful comments and suggestions. The main part of the work was done while both authors were at Siemens Corporate Research, Inc., Princeton, NJ, USA. In addition, Haibin Ling is partially supported by NSF grant (ITR-03258670325867).

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] D. Androutsos, K. N. Plataniotis and A. N. Venetsanopoulos. “A novel vector-based approach to color image retrieval using a vector angular-based distance measure”. *Computer Vision and Image Understanding*, 75(1-2):46-58, 1999.

[3] S. Belongie, J. Malik and J. Puzicha. “Shape Matching and Object Recognition Using Shape Context”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(24):509-522, 2002.

[4] S. Cohen, L. Guibas. “The Earth Mover’s Distance under Transformation Sets”, *IEEE International Conference on Computer Vision*, II:1076-1083, 1999.

[5] A. Darabiha, J. Rose and W. J. MacLean. “Video-Rate Stereo Depth Measurement on Programmable Hardware,”, *IEEE Conference on Computer Vision and Pattern Recognition*, I:203-210, 2003.

[6] W. Freeman, E. Pasztor, and O. Carmichael. “Learning low level vision”. *International Journal of Computer Vision*, 40(1):24-47, 2000.

[7] K. Grauman and T. Darrell, “Fast Contour Matching Using Approximate Earth Mover’s Distance”, *IEEE Conference on Computer Vision and Pattern Recognition*, I:220-227, 2004

[8] K. Grauman and T. Darrell. “The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features”. *IEEE International Conference on Computer Vision*, II:1458-1465, 2005.

[9] C. Grigorescu and N. Petkov. “Distance sets for shape filters and shape recognition”. *IEEE Trans. on Image Processing*, 12(10):1274-1286, 2003.

[10] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. "Efficient color histogram indexing for quadratic form distance functions", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(7):729-736, 1995.

[11] C. Harris and M. Stephens, "A combined corner and edge detector", *Alvey Vision Conference*, 147-151, 1988.

[12] F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming*. McGraw-Hill, New York, NY, 1990.

[13] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *Jour. Math. Phys.*, 20:224-230, 1941.

[14] A. S. Holmes, C. J. Rose, and C. J. Taylor. "Measuring similarity between pixel signatures," *Image and Vision Computing*, 20(5):331-340, 2002.

[15] A. S. Holmes, C. J. Rose, and C. J. Taylor. "Transforming pixel signatures into an improved metric space," *Image and Vision Computing*, 20(9):701-707, 2002.

[16] P. Indyk and N. Thaper, "Fast Image Retrieval via Embeddings", *In 3rd Workshop on Statistical and computational Theories of Vision*, Nice, France, 2003

[17] A. C. Jalba, M. H. F. Wilkinson and J. B. T. M. Roerdink. "Shape Representation and Recognition Through Morphological Curvature Scale Spaces". *IEEE Trans. on Image Processing*, 15(2):331-341, 2006.

[18] A. Johnson, M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433-449, 1999.

[19] D. G. Jones and J. Malik. "A computational framework for determining stereo correspondence from a set of linear spatial filters". *European Conference on Computer Vision*, 395-410, 1992.

[20] N. Karmarkar. "A New Polynomial-time Algorithm for Linear Programming". In *Proc. of the 16th Annual ACM Symp. on Theory of Computing*, 302-311, 1984.

[21] Y. Ke and R. Sukthankar. "PCA-SIFT: a more distinctive representation for local image descriptors", *IEEE Conference on Computer Vision and Pattern Recognition*, II:506-513, 2004.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using affine-invariant regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(8):1265-1278, 2005.

[23] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour", *IEEE Conference on Computer Vision and Pattern Recognition*, I:424-429, 2000.

[24] E. Levina and P. Bickel. "The Earth Mover's Distance is the Mallows Distance: Some Insights from Statistics", *IEEE International Conference on Computer Vision*, 251-256, 2001.

[25] J. Lin. "Divergence measures based on the Shannon entropy". *IEEE Trans. on Information Theory*, 37(1):145-151, 1991.

[26] H. Ling and D. W. Jacobs, "Using the Inner-Distance for Classification of Articulated Shapes", *IEEE Conference on Computer Vision and Pattern Recognition*, II:719-726, 2005.

[27] H. Ling and D. W. Jacobs, "Deformation Invariant Image Matching", *IEEE International Conference on Computer Vision*, II:1466-1473, 2005.

[28] H. Ling and K. Okada. "EMD-L_1: An Efficient and Robust Algorithm for Comparing Histogram-Based Descriptors", *European Conference on Computer Vision (ECCV)*, LNCS 3953, III:330-343, 2006.

[29] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, 60(2), pp. 91-110, 2004.

[30] C. L. Mallows. "A note on asymptotic joint normality". *Annals of Mathematical Statistics*, 43(2):508-515, 1972.

[31] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615-1630, 2005.

[32] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA", *IEEE Conference on Computer Vision and Pattern Recognition*, I:1063-6919, 2003.

[33] F. Mokhtarian, S. Abbasi and J. Kittler. "Efficient and Robust Retrieval by Shape Content through Curvature Scale Space," in A. W. M. Smeulders and R. Jain, editors, *Image Databases and Multi-Media Search*, 51-58, World Scientific, 1997.

[34] E. N. Mortensen, H. Deng, and L. Shapiro, "A SIFT Descriptor with Global Context", *IEEE Conference on Computer Vision and Pattern Recognition*, I:184-190, 2005.

[35] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Pektovic, P. Yanker, C. Faloutsos, and G. Taubin. "The QBIC project: querying images by content using color, texture and shape". *In Proc. of SPIE Storage and Retrieval for Image and Video Databases*, pp.173-187, 1993.

[36] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. "Shape Distributions", *ACM Trans. on Graphics*, 21(4):807-832, 2002.

[37] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[38] S. Peleg, M. Werman, and H. Rom. "A Unified Approach to the Change of Resolution: Space and Gray-level", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:739-742, 1989.

[39] S. T. Rachev. "The Monge-Kantorovich mass transference problem and its stochastic applications". *Theory of Probability and its Applications*, XXIX(4):647C676, 1984.

[40] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. "Empirical Evaluation of Dissimilarity Measures for Color and Texture", *Computer Vision and Image Understanding*, 84:25-43, 2001.

[41] Y. Rubner and C. Tomasi. *Perceptual Metrics for Image Database Navigation*. Kluwer Academic Publishers, Boston, MA, 2001.

[42] Y. Rubner, C. Tomasi, and L. J. Guibas. "The Earth Mover's Distance as a Metric for Image Retrieval", *International Journal of Computer Vision*, 40(2):99-121, 2000.

[43] T. B. Sebastian, P. N. Klein and B. B. Kimia. "On Aligning Curves", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(1):116-125, 2003.

[44] H. C. Shen and A. K. C. Wong. "Generalized texture representation and metric". *Computer Vision, Graphics, and Image Processing*, 23:187-206, 1983.

[45] M. J. Swain and D. H. Ballard. "Color Indexing", *International Journal of Computer Vision*, 7(1):11-32, 1991.

[46] H. Tan and C. Ngo. "Common Pattern Discovery Using Earth Movers Distance and Local Flow Maximization", *IEEE International Conference on Computer Vision*, II:1222-1229, 2005.

[47] A. Thayananthan, B. Stenger, P. H. S. Torr and R. Cipolla, "Shape Context and Chamfer Matching in Cluttered Scenes", *IEEE Conference on Computer Vision and Pattern Recognition*, I:127-133, 2003.

[48] Z. Tu and A. L. Yuille. "Shape Matching and Recognition-Using Generative Models and Informative Features", *European Conference on Computer Vision*, III:195-209, 2004.

[49] M. Werman, S. Peleg, and A. Rosenfeld. "A distance metric for multidimensional histograms". *Computer Vision, Graphics, and Image Processing*, 32:328-336, 1985.

[50] G. Wesolowsky. "The Weber Problem: History and Perspectives", *Location Science*, 1(1):5-23, 1993.

[51] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines". *Neural Information Processing Systems*, 16, 2003.