EMD-L_1: An Efficient and Robust Algorithm for Comparing Histogram-Based Descriptors

Haibin Ling[1] and Kazunori Okada[2]

[1] Computer Science Dept., Center for Automation Research, University of Maryland,
College Park, Maryland 20770, USA
hbling@umiacs.umd.edu

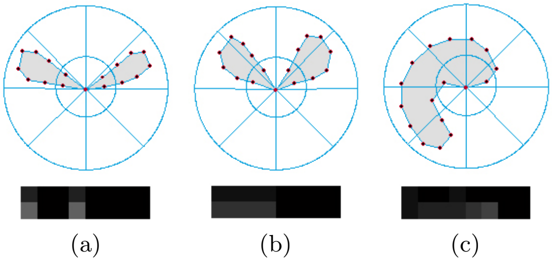
[2] Imaging and Visualization Department, Siemens Corporate Research, Inc.
755 College Rd. E. Princeton, New Jersey, 08540, USA
kazunori.okada@siemens.com

Abstract. We propose a fast algorithm, EMD-L_1, for computing the Earth Mover's Distance (EMD) between a pair of histograms. Compared to the original formulation, EMD-L_1 has a largely simplified structure. The number of unknown variables in EMD-L_1 is $O(N)$ that is significantly less than $O(N^2)$ of the original EMD for a histogram with N bins. In addition, the number of constraints is reduced by half and the objective function is also simplified. We prove that the EMD-L_1 is formally equivalent to the original EMD with L_1 ground distance without approximation. Exploiting the L_1 metric structure, an efficient tree-based algorithm is designed to solve the EMD-L_1 computation. An empirical study demonstrates that the new algorithm has the time complexity of $O(N^2)$, which is much faster than previously reported algorithms with super-cubic complexities. The proposed algorithm thus allows the EMD to be applied for comparing histogram-based features, which is practically impossible with previous algorithms. We conducted experiments for shape recognition and interest point matching. EMD-L_1 is applied to compare shape contexts on the widely tested MPEG7 shape dataset and SIFT image descriptors on a set of images with large deformation, illumination change and heavy noise. The results show that our EMD-L_1-based solutions outperform previously reported state-of-the-art features and distance measures in solving the two tasks.

1 Introduction

Histogram-based descriptors are used widely in various computer vision tasks such as shape matching [1, 22, 23, 13], image retrieval [15, 8, 18, 16], texture analysis [19, 9]. For comparing these descriptors, *bin-to-bin* distance functions, such as L_p distance, χ^2 statistics, and KL divergence, are most commonly used. These approaches assume that the domain of the histograms are already aligned. However, in practice, such assumption can be violated due to various factors, such as shape deformation, lighting variation, heavy noise, etc. The *Earth Mover's Distance* (EMD) [20] is a *cross-bin* dissimilarity function that addresses the above alignment problem by solving the *transportation problem* as a special case of

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part III, LNCS 3953, pp. 330–343, 2006.



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

(d)

Fig. 1. An example where bin-to-bin distances meet problems. (a), (b) and (c) show three shapes with log-polar bins on them and corresponding shape context histograms. (d) lists the distances between them using different distance functions.

linear programming (LP). Beyond the color signature application proposed by Rubner et al. [20] originally, we claim that EMD is useful for more general class of histogram descriptors such as SIFT [15] and shape context [1].

Fig. 1 shows an example with shape context [1]. 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 closer to (c) than to (a). Despite this favorable robustness property, EMD has seldom been applied to general histogram-based local descriptors to our knowledge. The main reason lies in its expensive computational cost, which is super-cubic[1] for a histogram with N bins.

Rubner et al. [20] proposed using the *transportation simplex* (TS) [6] to solve the EMD. They showed that TS has a super-cubic average time complexity. In [20], EMD is applied to compact signatures instead of raw distributions directly. This approach is efficient and effective especially for distributions with sparse structures, e.g., color histograms in the CIE-Lab space in [20]. However, the histogram-based descriptor is generally not sparse and can not be modelled compactly. This forces the EMD algorithm to be applied to the raw distribution directly. In real vision problems, the number of comparisons between these descriptors is very large, which forbids the use of TS algorithm. Cohen and Guibas [2] studied the problem of computing a transformation between distributions with minimum EMD. Levina and Bickel [11] proved that EMD is equivalent to the Mallows distance when applied to probability distributions. The L_1 formulation had been introduced by Wesolowsky [24] and then Cohen and Guibas [2]. In this paper we extended it to general histograms.

Indyk and Thaper [7] proposed a fast algorithm for image retrieval by embedding the EMD metric into a Euclidean space. Grauman and Darrell [3] extended the approach for contour matching. The embedding is performed using a hierarchical distribution analysis. A fast nearest neighbor retrieval is achieved through locality-sensitive hashing. EMD can be approximated by measuring the L_1 distance in the Euclidean space after embedding. The time complexity of the embedding is $O(Nd \log \Delta)$, where N is the size of feature sets, d is the dimension of the feature space and Δ is the diameter of the union of the two feature sets to

[1] By super-cubic, we mean a complexity between $O(N^3)$ and $O(N^4)$

be compared. These approaches are efficient for retrieval tasks and global shape comparison [7, 3]. However, they focused on the feature set matching rather than the histogram comparison of our interest. In addition, they are approximative. Thus the errors introduced by the embedding may reduce the performance for the histogram-based descriptors. Recently, Grauman and Darrell [4] proposed using the *pyramid matching kernel* (PMK) for feature set matching. PMK further 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 the partial matching through histogram intersections.

The contribution of this paper is twofold. First, 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 formulation. It has only $O(N)$ unknown variables, which is 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. Unlike previous approximative algorithms, we formally prove that EMD-L_1 is equivalent to the original EMD with L_1 ground distance. An efficient tree-based algorithm is designed to solve EMD-L_1 and an empirical study shows that the time complexity of EMD-L_1 is $O(N^2)$, which significantly improves the previous super-cubic algorithm.

Second, the speedup gained by EMD-L_1 enables us to compute the exact EMD directly for histograms without reducing the discriminability. For the first time, EMD is applied to compare histogram-based local descriptors. We tested EMD-L_1 in two experiments. First, it is applied to the inner-distance shape context [13] for shape matching on the widely tested MPEG7 shape dataset, where EMD-L_1 achieves a better score than all previously reported results. Second, EMD-L_1 is applied to the SIFT [15] descriptors for feature matching on images with large distortion. Again, EMD-L_1 demonstrates excellent performance. In addition, it also shows that EMD-L_1 performs similar to the original EMD with L_2 ground distance, while the latter is much slower.

The rest of the paper is organized as follows. Sec. 2 reviews the EMD and derives its formulation for histograms. Sec. 3 first gives the formulation of EMD-L_1. Then, the equivalence between EMD-L_1 and EMD with L_1 ground distance is proved. Finally a fast algorithm for EMD-L_1 is proposed, followed by an empirical study of time complexity. Sec. 4 describes the experiments of applying the EMD-L_1 to shape recognition and interest point matching. Sec. 5 concludes.

2 The Earth Mover's Distance (EMD)

2.1 The EMD Between Signatures

The Earth Mover's Distance (EMD) is proposed by Rubner et al. [20] to measure the dissimilarity between *signatures*. Signatures are extracted from distributions via clustering. 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 *transportation problem*.

The elements in P are treated as "supplies" located at u_i's and element in Q as "demands" at v_j's. p_i and q_j indicate the amount of supply and demand respectively. The EMD is the minimum (normalized) work required for this task. It is defined as

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

such that $\sum_j f_{ij} \leq p_i$, $\sum_i f_{ij} \leq q_j$, $\sum_{i,j} f_{ij} = \min\{\sum_i p_i, \sum_j q_j\}$ and $f_{ij} \geq 0$. $F = \{f_{ij}\}$ is the set of *flows*. f_{ij} represents the amount transported from the i-th supply to the j-th demand. d_{ij} is a distance between the position u_i and v_j called the *ground distance*.

2.2 The EMD Between Histograms

Histograms can be viewed as a special type of signatures in that each bin corresponding to an element in a signature. Specifically, 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 will discuss two dimensional histograms which are widely used for shape and image descriptors. Higher dimensional cases can be derived similarly. Wlog, we use the following assumptions and notations.

- The histogram has m rows and n columns and $N = m \times n$ bins.
- The index set for bins is defined as $\mathcal{I} = \{(i,j) : 1 \leq i \leq m, 1 \leq j \leq n\}$. 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 1, i.e., $\sum_{i,j} p_{ij} = 1$, $\sum_{i,j} q_{ij} = 1$.

Now the EMD between two histograms P and Q becomes

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

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

Where F is the flow from P to Q, i.e., $f_{i,j;k,l}$ is a flow from bin (i,j) to (k,l). Note that we use "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 usually defined by L_p distance

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

3 EMD-L_1

This section presents the EMD-L_1, a more efficient formulation of the EMD between histograms. We first show that, by using L_1 or Manhattan distance as the ground distance, the EMD-L_1 is drastically simplified compared to the original one. Then, we prove that EMD-L_1 is equivalent to the original EMD with L_1 ground distance. Finally an efficient algorithm and an empirical complexity study are presented.

3.1 EMD with L_1 Ground Distance

As shown later in Sec. 4.2 and Fig. 7(b), EMD's with L_1 and L_2 ground distances performs similarly for our purpose, while the former is much faster. Therefore, we are interested in L_1 ground distance. In the rest of the paper, L_1 ground distance is implicitly assumed. With L_1 ground distance, formula (3) becomes

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

Note that the ground distance now takes only integer values. For convenience of discussion, the flow index set $\mathcal{J}$ is divided into three disjointed parts $\mathcal{J} = \mathcal{J}_0 \bigcup \mathcal{J}_1 \bigcup \mathcal{J}_2$, each of them corresponds to one of three flow types.

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

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

$$d_{i,j;k,l} = d_{i,j;i,l} + d_{i,l;k,l} = \sum\nolimits_{j\le x<l} d_{i,x;i,x+1} + \sum\nolimits_{i\le y<k} d_{y,l;y+1,l} \ .$$

Accordingly, the shortest path from (i,j) to (k,l) can be decomposed into neighbor edges. It follows that, without changing the total weighted flow $\sum_{f\in F} fd$, $f_{i,j;k,l}$ can be set to zero by first increasing all n-flows along the path $[(i,j),(i,j+1),\ldots,(i,l),(i+1,l),\ldots,(k,l)]$ by $f_{i,j;k,l}$. This is illustrated in Fig. 2

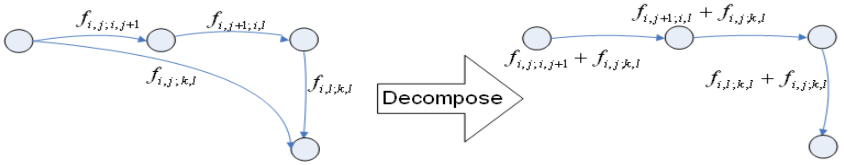

$f_{i,j;i,j+1}$
$f_{i,j+1;i,l}$
$f_{i,j;k,l}$
$f_{i,l;k,l}$
Decompose
$f_{i,j+1;i,l} + f_{i,j;k,l}$
$f_{i,j;i,j+1} + f_{i,j;k,l}$
$f_{i,l;k,l} + f_{i,j;k,l}$

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

S-flows are also redundant due to their zero ground distances. With these intuitions, we propose a new formulation of EMD, *EMD-L_1*, as below

$$EMD_{L_1}(P,Q) = \min_{G=\{g_{i,j;k,l}:(i,j,k,l)\in\mathcal{J}_1\}} \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{4}$$

$$\text{s.t.} \begin{cases} \sum_{k,l:(i,j,k,l)\in\mathcal{J}_1} (g_{i,j;k,l} - g_{k,l;i,j}) = b_{ij} & \forall (i,j) \in \mathcal{I} \\ g_{i,j;k,l} \geq 0 & \forall (i,j,k,l) \in \mathcal{J}_1 \end{cases} \tag{5}$$

Where $b_{ij} = p_{ij} - q_{ij}$ is the difference between the two histograms. We call a flow G satisfying (5) a *feasible* flow analogous to that in the original EMD.

EMD-L_1 has large simplifications over the original EMD (1), including

1. There are only about $O(N)$ variables in (4), 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 [6]. In addition, the space efficiency gained by this is very favorable for large histograms.
2. The number of equality constraints is reduced by half. This is another important factor for the efficiency of the LP algorithms.
3. All the ground distances involved in the EMD-L_1 are ones. This is practically useful, because it saves multiplications during computation and allows the use of integer operations to handle the coefficients.

Note that these simplifications can be extended to higher dimensional cases. For example, the unknown variables for 3D histograms is $6N$ thus still of $O(N)$ complexity. These simplifications are used to design a fast tree-based algorithm.

3.2 Equivalence Between EMD-L_1 and Original EMD

We now prove the equivalence between the EMD-L_1 and the original EMD with L_1 ground distance. The equivalence is in the sense of the weighted total flows. That is, 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

$$EMD(P,Q) = EMD_{L_1}(P,Q) \; . \tag{6}$$

The discussion in the last subsection hints that, for any flow F for the original EMD, an equivalent flow G for EMD-L_1 can be created by eliminating f-flows and s-flows. This implies $EMD(P,Q) \geq EMD_{L_1}(P,Q)$. 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 use a “merge” procedure instead of the decomposition. The idea is to merge input and output flows at each bin such that either input or output flow survives as a result. This is demonstrated in Fig. 3. Notice that we only need an F to have a total weight not greater than that of G. This makes the merge procedure much simpler, since we can just merge any pair of input and output flows.

Proof. It suffices to prove
$EMD(P,Q) \geq EMD_{L_1}(P,Q)$ and $EMD(P,Q) \leq EMD_{L_1}(P,Q)$.

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

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.

$$\sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} = \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{7}$$

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

$$\begin{cases} \sum_{\mathcal{J}} f'_{i,j;k,l} d_{i,j;k,l} = \sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} & \\ \sum_{k,l} (f'_{i,j;k,l} - f'_{k,l;i,j}) = b_{ij} & \forall (i,j) \in \mathcal{I} \\ f'_{i,j;k,l} \geq 0 & \forall (i,j,k,l) \in \mathcal{J} \end{cases} \tag{8}$$

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. 2. In detail, assume $i \leq k, j \leq l$ (other cases are similar), the three modifications to F' are conducted in the given order

$$\begin{cases} f'_{i,x;i,x+1} \leftarrow f'_{i,x;i,x+1} + f'_{i,j;k,l} & \forall x, j \leq x < l \\ f'_{y,l;y+1,l} \leftarrow f'_{y,l;y+1,l} + f'_{i,j;k,l} & \forall y, i \leq y < k \\ f'_{i,j;k,l} \leftarrow 0 & \end{cases} \tag{9}$$

It is clear that (8) always holds because (9) does not change it. After all the f-flows vanish, we build G from F'

$$g_{i,j;k,l} = f'_{i,j;k,l} \ , \ \forall (i,j,k,l) \in \mathcal{J}_1 \tag{10}$$

From (8) it follows that G satisfies (5) and (7).

Part II. Proof of $EMD(P,Q) \leq EMD_{L_1}(P,Q)$.

It suffices to prove that, for any $G = \{g_{i,j;k,l} : (i,j,k,l) \in \mathcal{J}_1\}$ satisfying (5), there exists $F = \{f_{i,j;k,l} : (i,j,k,l) \in \mathcal{J}\}$ satisfying (2), such that

$$\sum_{\mathcal{J}} f_{i,j;k,l} d_{i,j;k,l} \leq \sum_{\mathcal{J}_1} g_{i,j;k,l} \tag{11}$$

For any G satisfying (5), we create an auxiliary flow $G' = \{g'_{i,j;k,l} : (i,j,k,l) \in \mathcal{J}\}$. G' is first initialized by G

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

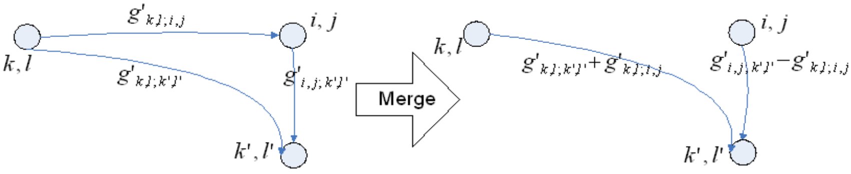


Fig. 3. Flow merging, where $b_{ij} > 0$, $g'_{i,j;k',l'} > g'_{k,l;i,j} > 0$

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

$$\begin{cases} \sum_{\mathcal{J}} g'_{i,j;k,l} d_{i,j;k,l} \leq \sum_{\mathcal{J}_1} g_{i,j;k,l} & \\ \sum_{k,l \in \mathcal{I}} (g'_{i,j;k,l} - g'_{k,l;i,j}) = b_{ij} & \forall (i,j) \in \mathcal{I} \\ g'_{i,j;k,l} \geq 0 & \forall (i,j,k,l) \in \mathcal{J} \end{cases} \tag{12}$$

Note that in the first equation of (12) "$\leq$" is used instead of "$=$".

Now we will 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
      $$\begin{cases} \delta \leftarrow \min\{g'_{i,j;k',l'}, g'_{k,l;i,j}\} \\ g'_{k,l;k',l'} \leftarrow g'_{k,l;k',l'} + \delta \\ g'_{k,l;i,j} \leftarrow g'_{k,l;i,j} - \delta \\ g'_{i,j;k',l'} \leftarrow g'_{i,j;k',l'} - \delta \end{cases} \tag{13}$$
    END WHILE
  END FOR
```

Fig. 3 shows an example of merging. The four steps in (13) need to be applied in the order as given. Moreover, each run of (13) removes at least one non-zero flow, so the procedure is guaranteed to terminate.

Because of the triangle inequality $d_{k,l;k',l'} \leq d_{k,l;i,j} + d_{i,j;k',l'}$, (13) will only decrease the left hand side of the first inequality in (12) and hence will not change it. The second equation in (12) also holds because (13) changes the input and output flows of a node with the same amount (δ). The third condition in (12) is obvious.

An important observation due to (12) and the procedure is

$$\begin{cases} g'_{i,j;k,l} = 0 & \forall (i,j,k,l) \in \mathcal{J} \text{ if } b_{ij} \leq 0 \\ g'_{k,l;i,j} = 0 & \forall (i,j,k,l) \in \mathcal{J} \text{ if } b_{ij} \geq 0 \end{cases} \tag{14}$$

Now we build F from G':

$$f_{i,j;k,l} = \begin{cases} \min\{p_{ij}, q_{kl}\} & \forall (i,j,k,l) \in \mathcal{J}_0 \\ g'_{i,j;k,l} & \forall (i,j,k,l) \in \mathcal{J}_1 \bigcup \mathcal{J}_2 \end{cases} \tag{15}$$

From (14), (12) and (15), we have that F satisfies (2) and (11). ■

3.3 Algorithmic Solution for EMD-L_1

EMD-L_1 is clearly a LP problem by its definition. The simplex algorithm becomes a natural solution. In addition, EMD-L_1 also has a very special structure similar to the original EMD. Therefore, a fast simplex algorithm can be designed analogous to the transportation simplex used for the original EMD [20, 6]. We propose an even faster tree-based algorithm, *Tree-EMD*. The algorithm can be derived from the fast simplex algorithm. It takes the benefit of the simplex while exploiting a tree structure for further speedup. In addition, Tree-EMD has a more intuitive interpretation. Finally, the tree structure also makes coding easy for different dimensions. Due to the space limitation, we only briefly describe the outlines of the algorithm and left the details to its longer version [14].

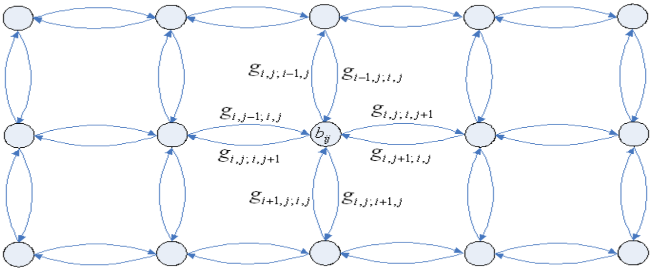


Fig. 4. The EMD-L_1 as a network flow problem for 3×5 histograms

To gain intuition, EMD-L_1 is modeled as a graph $\mathcal{G} =< V, B, G >$ as illustrated in Fig. 4. $V = \{v_{ij} : (i, j) \in \mathcal{I}\}$ is the set of nodes in the graph. $B = \{b_{ij} : (i, j) \in \mathcal{I}\}$ is the weights associated to V, $\sum_{\mathcal{I}} b_{ij} = 0$. G is the set of flows between neighbor nodes. The task is to find the minimum flow such that all nodes have zero weights after applying the flow.

Before describing Tree-EMD, we give some definitions derived from the classic simplex algorithm [6]. A flow G is called *feasible* if (5) holds. A feasible flow G is called a *basic feasible tree* (BFT) if G has only $mn-1$ elements that can be non-zero and they form a tree. Such elements are called *basic variable*(BV) flows.

Notes: 1) The loops and trees in this paper are undirected, although flows do have directions. 2) A BFT is actually a *spanning tree* since there are mn nodes in the graph.

The task becomes to find a feasible flow G with minimum total flow $\sum_{g \in G} g$. It can be shown that there exists an optimal BFT G [14]. Therefore, the search space of the optimum solution can be restricted within the set of BFT's.

Tree-EMD is an iterative algorithm for searching the optimum BFT tree. First, an initial BFT tree G is built using the greedy algorithm. Then, G is iteratively replaced by a better BFT tree with smaller flow until the optimum is reached. In each iteration, an *entering* flow $g_{i_0,j_0;k_0,l_0}$ is found and added to G. Accordingly, a *leaving* BV flow $g_{i_1,j_1;k_1,l_1}$ is picked to avoid loops. G is then modified by adding $g_{i_0,j_0;k_0,l_0}$ and removing $g_{i_1,j_1;k_1,l_1}$ and adjusting flow values to keep it as a BFT. The iteration is guaranteed to terminate at a global minimum due to its underlying simplex algorithm.

The most important variables in Tree-EMD are u_{ij}'s for nodes v_{ij}'s and $c_{i,j;k,l}$ for flows $g_{i,j;k,l}$'s. They have following relations

$$c_{i,j;k,l} = 1 - u_{i,j} + u_{k,l} \qquad \forall (i,j,k,l) \in \mathcal{J}_1 \tag{16}$$

$$c_{i,j;k,l} = 1 - u_{i,j} + u_{k,l} = 0 \quad \text{if } g_{i,j;k,l} \text{ is a BV flow} \tag{17}$$

We now discuss several key issues in the algorithm.

1. *Optimality test*: A BFT G is optimum iff $c_{i,j;k,l} \geq 0$, $\forall (i,j,k,l) \in \mathcal{J}_1$.
2. *Finding* $g_{i_0,j_0;k_0,l_0}$: $(i_0, j_0, k_0, l_0) = \text{argmin}_{(i,j,k,l)\in\mathcal{J}_1} c_{i,j;k,l}$.
3. *Finding* $g_{i_1,j_1;k_1,l_1}$: First, find the loop formed by adding $g_{i_0,j_0;k_0,l_0}$ into G. Then $g_{i_1,j_1;k_1,l_1}$ is the flow in the loop with minimum flow value and reversed direction of $g_{i_0,j_0;k_0,l_0}$.
4. *Updating* G: First, adding $g_{i_0,j_0;k_0,l_0}$ in G. Then modify flow values along the loop mentioned above ($g_{i_1,j_1;k_1,l_1}$ becomes zero). After that, remove $g_{i_1,j_1;k_1,l_1}$ and adjust the links in G accordingly.

Table 1. Tree-EMD

```
Step 0 /*Define some key variables*/
         $r$: the root of the tree
         $p^*$: the root of the subtree to be updated
Step 1 /*Initialization*/
         Initialize BFT by a greedy initial solution
         $p^* \leftarrow r$
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 (17)
                  Recursively update $q$'s children
              END FOR
              /*Optimality test*/
              Compute $c_{i,j;k,l}$'s
              IF (optimum is reached)    goto Step 3    END IF
              /*Find a new improved BF solution*/
              Find entering BV flow $g_{i_0,j_0;k_0,l_0}$
              Find loop by tracing from $v_{i_0,j_0}$ and $v_{k_0,l_0}$ to their common ancestor
              Find the leaving BV $g_{i_1,j_1;k_1,l_1}$
              Update flow values in $G$ along the loop
              Maintain the tree, include removing $g_{i_1,j_1;k_1,l_1}$, adding $g_{i_0,j_0;k_0,l_0}$
                  and updating links.
              Set $p^*$ as the root of subtree where $u_{ij}$'s need to be updated.
         END WHILE
Step 3 Compute the total flow as the EMD distance.
```

5. *Updating u_{ij}'s*: Fix u_{ij} of the root to zero. Other u_{ij}'s can be computed starting from the root by using (17). In fact, only a small amount of u_{ij}'s in a subtree need to be updated in more iterations.
6. *Updating $c_{i,j;k,l}$'s*: When u_{ij}'s determined, use formula (16).

A brief description of Tree-EMD is given in Table 1.

3.4 Empirical Study for Time Complexity

To study the time complexity of the proposed algorithm, we conduct an empirical study similar to that in [20]. First, two sets of 2D random histograms for each size $n \times n$, $2 \leq n \leq 20$ are generated. For each n, 1000 random histograms are generated for each set. 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 TS[2]). In addition, EMD-L_1 is tested for 3D histograms with similar settings, except $2 \leq n \leq 8$. The results are shown in Fig. 5. From (a) it is clear that EMD-L_1 is much faster than the original one. (b) shows that EMD-L_1 has a complexity around $O(N^2)$, where N is the number of bins (n^2 for 2D and n^3 for 3D).

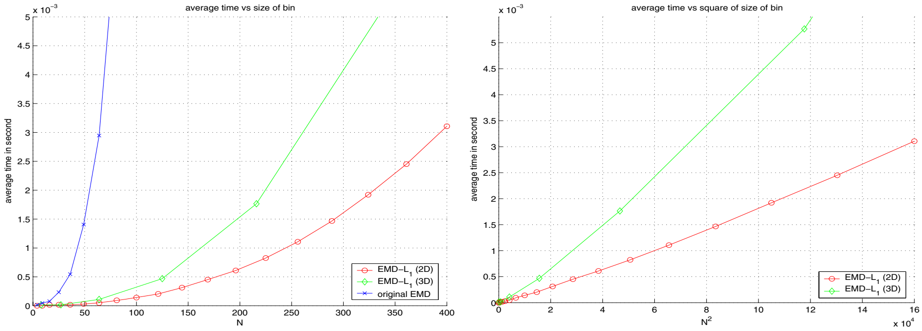


Fig. 5. Empirical time complexity study of EMD-L_1 (Tree-EMD). Left: In comparison to the original EMD (TS). Right: Average running time vs. square of histogram sizes.

4 Experiments

4.1 Shape Matching with Shape Context

EMD-L_1 is tested for shape matching by applying it to the inner-distance shape context (IDSC)[13]. IDSC is an extension of shape context (SC)[1] by using the shortest path distances. These studies used χ^2 distance for comparing the shape descriptors. In [13], IDSC is used for contour comparison with a dynamic programming (DP) scheme. We use the same framework, except for replacing the χ^2 distance with the EMD-L_1. In addition, the lower bound of EMD [20] is used for speeding up the dynamic programming.

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

Fig. 6. Typical shape images from the MPEG7 CE-Shape-1, one image per class

Table 2. Retrieval rate (bullseye) of different methods for the MPEG7 CE-Shape-1

Alg.	CSS[17]	Vis. Parts[10]	SC[1]	Curve Edit[21]	Gen. Mod.[23]	IDSC[13]	EMD-L_1
Score	75.44%	76.45%	76.51%	78.17%	80.03%	85.40%	86.56%

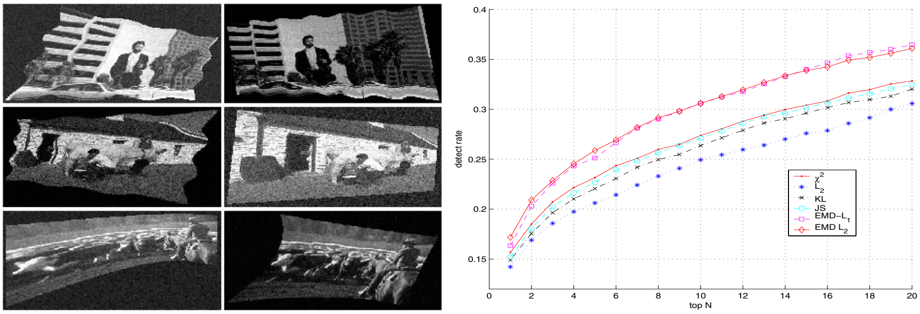
The MPEG7 CE-Shape-1 [10] database is widely used for benchmarking different shape matching algorithms. The data set contains 1400 silhouette images from 70 classes. Each class has 20 different shapes (e.g. Fig. 6). The performance is measured by the Bullseye test. Every image in the database is matched 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 highest possible number of hits (20x1400).

We use the same experimental setup as [13]. The bullseye score is listed in Tab. 2 with previously reported results. The excellent performance, outperforming the previous best scores, demonstrates the effectiveness of EMD-L_1.

4.2 Image Feature Matching

This subsection describes our experiment using the EMD-L_1 for interest point matching. The experiment was conducted on a set of ten image pairs containing synthetic deformation, noise and illumination change. Some testing images are shown in Fig. 7 (a).

Interest point. We use Harris corners [5] for the matching experiments. The reason for this choice is that, due to the large deformation, noise and lighting change, it is hard to apply other interest point detectors. Furthermore, we focus more on comparing descriptors than the interest points. For each image, we pick 300 points with the largest cornerness responses.



(a) (b)

Fig. 7. (a) Some testing images with synthetic deformation, illumination change and noise. (b) ROCs for EMD-L_1 and other dissimilarity functions on SIFT.

Descriptors. We use the SIFT proposed by Lowe [15] as the descriptors. SIFT is a very popular histogram-based descriptor. In our case, since scale invariant detectors are not available, a fixed support region is used (with diameter 41, similar to the setting used in [16]. SIFT is a three dimensional weighted histogram, 4 for each spatial dimensions and 8 for gradient orientation.

Evaluation criterion. For each pair of images together with their interest points, we first automatically obtained the ground truth correspondence from the synthesis procedure. Then, every interest point in Image 1 is compared with all interest points in Image 2 by comparing the SIFT extracted on them. An interest point p_1 in Image 1 is treated as a correct match of another point p_2 in Image 2 if the displacement of p_1 is within a fixed distance of p_2. The detection rate among the top N matches is used to study the performance. The detection rate is defined as: $r = \frac{\text{\# correct matches}}{\text{\# possible matches}} = \frac{\text{\# correct matches}}{\text{\# points in Image 1}}$.

Experiment results. We tested the EMD-L_1 along with several bin-to-bin distance measures, including χ^2, KL-divergence (symmetric), Jensen-Shannon(JS) divergence [12], L_2, etc. The EMD with L_2 ground distance is also tested for comparison. A Receiver Operating Characteristic (ROC) based criterion is used to show the detection rates versus N, which is the number of most similar matches allowed. The ROC curves for the experiment are shown in Fig. 7 (b). The EMD-L_1 outperforms all other bin-to-bin metrics. In addition, EMD-L_1 and EMD with L_2 ground distance have very similar performance, though the former takes about 25 seconds per pair while the latter takes about 2100 seconds.

5 Conclusion

We propose a fast algorithm, EMD-L_1 for computing Earth Mover's Distance (EMD) between histograms with L_1 ground distance. The new algorithm reformulates the EMD into a drastically simplified version by using the special structure of L_1 metric on histograms. We proved that EMD-L_1 is equivalent to the EMD with L_1 ground distance for histograms. We then designed an efficient tree-based algorithm to solve the EMD-L_1. An empirical study shows that EMD-L_1 is significantly faster than previous EMD algorithms. The speedup allows the EMD to be applied to 2D/3D histogram-based features for the first time. Experiments on both shape descriptors (shape context [1]) and image features (SIFT [15]) show the superiority of EMD-L_1 for handling the matching tasks with large deformation, noise and lighting change, etc.

Acknowledgments

We would like to thank David Jacobs, Leo Grady, and Kevin Zhou for stimulating discussions. We would also thank Yossi Rubner for the TS code and anonymous reviewers for valuable comments. The work was mainly done during Haibin Ling's internship at Siemens Corporate Research. Haibin Ling also thank NSF grant (ITR-03258670325867) for supporting.

References

1. S. Belongie, J. Malik and J. Puzicha. “Shape Matching and Object Recognition Using Shape Context”, *IEEE Trans. on PAMI*, 24(24):509-522, 2002.
2. S. Cohen, L. Guibas. “The Earth Mover’s Distance under Transformation Sets”, *ICCV*, II:1076-1083, 1999.
3. K. Grauman and T. Darrell, “Fast Contour Matching Using Approximate Earth Mover’s Distance”, *CVPR*, I:220-227, 2004
4. K. Grauman and T. Darrell. “The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features”. *ICCV*, II:1458-1465, 2005.
5. C. Harris and M. Stephens, “A combined corner and edge detector”, *Alvey Vision Conference*, 147-151, 1988.
6. F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming.* McGraw-Hill, New York, NY, 1990.
7. P. Indyk and N. Thaper, “Fast Image Retrieval via Embeddings”, *In 3rd Workshop on Statistical and computational Theories of Vision*, Nice, France, 2003
8. Y. Ke and R. Sukthankar. “PCA-SIFT: a more distinctive representation for local image descriptors”, *CVPR*, II:506-513, 2004.
9. S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using affine-invariant regions,” *IEEE Trans. PAMI*, 27(8):1265-1278, 2005.
10. L. J. Latecki, R. Lakamper, and U. Eckhardt, “Shape Descriptors for Non-rigid Shapes with a Single Closed Contour”, *CVPR*, I:424-429, 2000.
11. E. Levina and P. Bickel. “The Earth Mover’s Distance is the Mallows Distance: Some Insights from Statistics”, *ICCV*, 251-256, 2001.
12. J. Lin. “Divergence measures based on the Shannon entropy”. *IEEE Trans. on Information Theory*, 37(1):145-151, 1991.
13. H. Ling and D. W. Jacobs, “Using the Inner-Distance for Classification of Articulated Shapes”, *CVPR*, II:719-726, 2005.
14. H. Ling and K. Okada. “An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison”, in submission, 2006.
15. D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *IJCV*, 60(2), pp. 91-110, 2004.
16. K. Mikolajczyk and C. Schmid, “A Performance Evaluation of Local Descriptors,” *IEEE Trans. on PAMI*, 27(10):1615-1630, 2005.
17. F. Mokhtarian, S. Abbasi and J. Kittler. “Efficient and Robust Retrieval by Shape Content through Curvature Scale Space,” *Image Databases and Multi-Media Search*, 51-58, World Scientific, 1997.
18. E. N. Mortensen, H. Deng, and L. Shapiro, “A SIFT Descriptor with Global Context”, *CVPR*, I:184-190, 2005.
19. S. Peleg, M. Werman, and H. Rom. “A Unified Approach to the Change of Resolution: Space and Gray-level”, *IEEE Trans. on PAMI*, 11:739-742, 1989.
20. Y. Rubner, C. Tomasi, and L. J. Guibas. “The Earth Mover’s Distance as a Metric for Image Retrieval”, *IJCV*, 40(2):99-121, 2000.
21. T. B. Sebastian, P. N. Klein and B. B. Kimia. “On Aligning Curves”, *IEEE Trans. on PAMI*, 25(1):116-125, 2003.
22. A. Thayananthan, B. Stenger, P. H. S. Torr and R. Cipolla, “Shape Context and Chamfer Matching in Cluttered Scenes”, *CVPR*, I:127-133, 2003.
23. Z. Tu and A. L. Yuille. “Shape Matching and Recognition-Using Generative Models and Informative Features”, *ECCV*, III:195-209, 2004.
24. G. Wesolowsky. “The Weber Problem: History and Perspectives”, *Location Science*, 1(1):5-23, 1993.