October 1997

Report No. STAN-CS-TR-97-1597

The Earth Mover's Distance: Lower Bounds and Invariance under Translation

by

Scott Cohen and Leonidas Guibas

Department of Computer Science

Stanford University
Stanford, California 94305

SF 298 MASTER COPY . KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 15, 1997

3. REPORT TYPE AND DATES COVERED
technical report

4. TITLE AND SUBTITLE
The Earth Mover's Distance: Lower Bounds and
Invariance under Translation

5. FUNDING NUMBERS
DAAH04-94-G-0284

6. AUTHOR(S)
Scott Cohen and Leonidas J. Guibas

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Department of Computer Science
Stanford University
Stanford, CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER
STAN-CS-TR-97-1597

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 33583.4-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The *Earth Mover's Distance* (EMD) between two finite distributions of weight is proportional to the minimum amount of work required to transform one distribution into the other. Current content-based retrieval work in the Stanford Vision Laboratory uses the EMD as a common framework for measuring image similarity with respect to color, texture, and shape content. In this report, we present some fast to compute lower bounds on the EMD which may allow a system to avoid exact, more expensive EMD computations during query processing. The effectiveness of the lower bounds is tested in a color-based retrieval system. In addition to the lower bound work, we also show how to compute the EMD under translation. In this problem, the points in one distribution are free to translate, and the goal is to find a translation that minimizes the EMD to the other distribution.

14. SUBJECT TERMS
content-based image retrieval, color-based image retrieval, earth mover's distance, lower bounds, translation invariance

15. NUMBER IF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Enclosure 1

The Earth Mover's Distance: Lower Bounds and Invariance under Translation [1]

Scott D. Cohen   Leonidas J. Guibas
(scohen, guibas)@cs.stanford.edu
Computer Science Department
Stanford University
Stanford, CA 94305

[1] This research was sponsored in part by the Defense Advanced Research Projects Agency under contract DAAH04-94-C-0284. This report is available online with color images through the Stanford University Computer Science Department Electronic Library at http://elib.stanford.edu/.

Abstract

The *Earth Mover's Distance* (EMD) between two finite distributions of weight is proportional to the minimum amount of work required to transform one distribution into the other. Current content-based retrieval work in the Stanford Vision Laboratory uses the EMD as a common framework for measuring image similarity with respect to color, texture, and shape content. In this report, we present some fast to compute lower bounds on the EMD which may allow a system to avoid exact, more expensive EMD computations during query processing. The effectiveness of the lower bounds is tested in a color-based retrieval system. In addition to the lower bound work, we also show how to compute the EMD under translation. In this problem, the points in one distribution are free to translate, and the goal is to find a translation that minimizes the EMD to the other distribution.

Contents

List of Figures

1 Introduction

Recent image-based retrieval work ([11, 12]) in the Stanford Vision Laboratory (SVL) has concentrated on providing a common framework for measuring image similarity with respect to color, texture, and shape content. In this framework, the summary or *signature* of an image is a finite collection of weighted points. For example, in [11] the color content signature of an image is a collection of dominant image colors represented in the CIE-Lab space, where each color is weighted by the fraction of image pixels classified as that color. In [12], the texture content signature of a single texture image is a collection of dominant spatial frequencies, where each frequency is weighted by the amount of energy at that frequency. In current shape-based retrieval work, the shape content signature of an image is a collection of points in parameter spaces of basic shapes (such as line segments and circular arcs) which fit well into image edges, where each basic shape occurrence is weighted by its length. To complete the uniform framework, a distance measure on weight distributions is needed to measure similarity between image signatures.

The *Earth Mover's Distance* (EMD) between two distributions is proportional to the minimum amount of *work* required to transform one distribution into the other. Here one unit of work is defined as the amount of work necessary to move one unit of weight by one unit of distance. The transformation process can be visualized as filling holes with piles of dirt. The holes are located at the points in the lighter distribution, and the dirt piles are located at the points in the heavier distribution. The volume of a hole or dirt pile is given by the weight value of its position. If the total weights of the distributions are equal, then all the dirt is used to fill the holes. Otherwise, there will be dirt leftover after all the holes have been completely filled. The EMD is defined to be the minimum amount of work to fill the holes divided by the total weight of the lighter distribution. Normalizing by the amount of dirt moved means the EMD will not change if the weights of both distributions are multiplied by a constant. The EMD is a metric when the total weights of the distributions are equal and the "ground distance" between holes and dirt piles is a metric ([12]). There is a very efficient method for computing the EMD which is based on a solution to the well-known *transportation problem* ([4]) in operations research.

In current SVL content-based retrieval systems, the distance between two images is taken as the EMD between the two corresponding signatures. The query time is dominated by the time to perform the EMD computations. Two common types of queries are nearest neighbor queries and range queries. In a nearest neighbor query, the system returns the K database images which are closest to the given query. In a range query, the system returns all database images which are within some distance r of the query. For both query types, fast lower bounds on the EMD may decrease the query time by avoiding slower, exact EMD computations. During nearest neighbor query processing, an exact EMD computation need not be performed if there is a lower bound on the EMD which is greater than the Kth smallest distance seen so far. During range query processing, an exact EMD computation need not be performed if there is a lower bound on the EMD which is greater than r. Of course, whether or not the query time decreases when a lower bound is used depends upon the number of exact EMD computations avoided and the computation times for the exact EMD and the lower bound.

It is known ([12]) that the distance between the centroids of two equal-weight distributions is a lower bound on the EMD between the distributions. There are, however, common situations in which distributions will have unequal weights. For example, consider the color-based retrieval work [11] in which the weight of a dominant image color is equal to the fraction of pixels classified as that color. Assuming all the pixels in an image are classified, the weight of every database signature is one. EMD comparisons between unequal-weight distributions arise whenever the system is presented with a *partial* query such as: "give me all images with at least 20% sky blue and 30% green". The query signature consists of two points in CIE-Lab space with weights equal to 0.20 and 0.30, and therefore has total weight equal to 0.50. In the texture world, it seems difficult to accurately classify every pixel in an image as one of a handful of dominant image textures. In this case, using the fraction of classified pixels as weight means that image distributions will have different weights. Of course, partial texture queries such as "give me all the images with at least 30% sand and 30% sky" also imply comparisons between distributions of unequal weight. In our current shape-based retrieval work, the weight of a basic shape that occurs in an image or illustration is equal to its length. Using length as weight, two image shape distributions are very likely to have different total weights. In all three cases, the total weight of a distribution is equal to the amount of information present in the underlying image. Since one cannot assume that all database images and queries will contain the same amount of information, lower bounds on the EMD between unequal-weight distributions may be quite useful in retrieval systems.

The first part of this report is dedicated to lower bounds on the EMD, and is organized as follows. In section 2, we give some basic definitions and notations that will be used thoughout the report. This section includes a formal definition of the Earth Mover's Distance. In section 3, we prove the centroid-distance lower bound for equal-weight distributions (section 3.1), and then we extend the idea behind this lower bound to obtain a centroid-based lower bound between unequal-weight distributions (section 3.2). In section 4, we present lower bounds which use projections of distribution points onto random lines through the origin and along the directions of the axes. These "projection-based" lower bounds involve the EMD between distributions on the real line, which is the subject of section 5. For one-dimensional distributions, we provide very efficient algorithms to compute (1) the EMD between equal-weight distributions and (2) a lower bound on the EMD between unequal-weight distributions. Both these algorithms use a single sweep over the distribution points. Furthermore, the lower bound for unequal weight case gives the exact EMD when applied in the equal weight case. In combination with the projection-based lower bounds in section 4, the exact and lower bound computations in one-dimension yield fast to compute lower bounds in general dimensions for both the equal and unequal-weight inputs. In section 6, we show some experiments that use our lower bounds in the previously mentioned color-based image retrieval system.

Another potentially useful area of exploration is computing the EMD under some given transformation group, such as the group of translations. In this problem, the points in one distribution can be transformed, and the goal is to find a transformation that minimizes the EMD to the other distribution. An application is shape-based retrieval, where visual similarity may not be captured by a direct comparison of the shapes present in two images due to differences in scale, orientation, and/or position. In the second part of this report, we consider the problem of computing the EMD under translation. In section 7, we give both a direct algorithm (section 7.1) and an iterative algorithm (section 7.2) for this problem. The direct algorithm is conceptually simple and is guaranteed to find a globally optimal translation, but it is not practical because it requires an unreasonable amount of time. The iterative method is efficient, but it may find only a locally optimal translation. Nonetheless, it may find a globally optimal translation if the iteration is run with a few different initial translations. Both algorithms require a subroutine that computes a point which minimizes the sum of weighted distances to a given set of points. This problem is the subject of section 8 where we give solutions when the distance function is the L_2-distance squared (section 8.1), the L_1-distance (section 8.2), and the Euclidean L_2-distance (section 8.3). Finally, in section 9, we give some concluding remarks on both EMD lower bounds and computing the EMD under a transformation group.

Note that the results presented in this report may still be very useful if one is interested in only the minimum work instead of the EMD, or one wants to use a different normalization factor than the weight of the lighter distribution. Statements about the EMD may be transformed into statements about the minimum work by multiplying through by the smaller weight. In fact, our reasoning about the EMD usually proceeds by reasoning about the work and dividing by the appropriate constant in the last step.

2 Basic Definitions and Notations

We denote a finite *distribution* x as

$$x = \{ \ (x_1, w_1), (x_2, w_2), \dots, (x_n, w_n) \ \} \equiv (X, w) \in \mathbf{D}^{d,n}$$

where

$$X = [\ x_1 \ \cdots \ x_n \ ] \in \mathbf{R}^{d \times n} \qquad \text{and} \qquad w \geq 0.$$

Here d is the dimension of the points $x_i \in \mathbf{R}^d$, and n is the number of points. For a vector v, let v_Σ be the sum of the components of v. The (total) *weight* of the distribution x is

$$w_\Sigma = \sum_{j=1}^{n} w_j.$$

Given two distributions $x = (X, w) \in \mathbf{D}^{d,m}$ and $y = (Y, u) \in \mathbf{D}^{d,n}$, a *flow* between x and y is any matrix $F = (f_{ij}) \in \mathbf{R}^{m \times n}$. Intuitively, f_{ij} represents the amount of weight at x_i which is matched to weight at y_j. An equally valid interpretation for f_{ij} is the amount of weight at y_j which is matched to weight at x_i. The term *flow* is meant to evoke the image of weight flowing from the points in the heavier distribution to the points in the lighter distribution until all the weight in the lighter distribution has been covered. If one distribution is known to be heavier than the other, then we shall write that a flow is *from* the heavier distribution *to* the lighter distribution. The flow F is a *feasible flow* between x and y iff

$$\begin{aligned} f_{ij} &\geq 0 \qquad i = 1, \dots, m, \ j = 1, \dots, n, && (1) \\ \sum_{j=1}^{n} f_{ij} &\leq w_i \qquad i = 1, \dots, m, && (2) \\ \sum_{i=1}^{m} f_{ij} &\leq u_j \qquad j = 1, \dots, n, \quad \text{and} && (3) \\ \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} &= \min(w_\Sigma, u_\Sigma). && (4) \end{aligned}$$

Constraint (1) requires the amount of x_i matched to y_j to be non-negative. Constraint (2) ensures that the weight in y matched to x_i does not exceed w_i. Similarly, (3) ensures that the weight in x matched to y_j does not exceed u_j. Finally, constraint (4) forces the total amount of weight matched to be equal to the weight of the lighter distribution.

Let $\mathcal{F}(x, y)$ denote the set of all feasible flows between x and y. The work done by a feasible flow $F \in \mathcal{F}(x, y)$ in matching x and y is given by

$$\text{WORK}(F, x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij},$$

where

$$d_{ij} = d(x_i, y_j)$$

is the distance between x_i and y_j. Throughout most of this report we shall use the Euclidean distance $d(x_i, y_j) = ||x_i - y_j||_2$ as the *ground distance* d, and this choice should be assumed unless otherwise indicated. The *Earth Mover's Distance* $\text{EMD}(x, y)$ between x and y is the minimum amount of work to match x and y, normalized by the weight of the lighter distribution:

$$\text{EMD}(x,y) = \frac{\min_{F=(f_{ij})\in\mathcal{F}(x,y)} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\min(w_\Sigma, u_\Sigma)} = \frac{\min_{F=(f_{ij})\in\mathcal{F}(x,y)} \text{WORK}(F,x,y)}{\min(w_\Sigma, u_\Sigma)}. \tag{5}$$

The work minimization problem in the numerator of (5) is a linear program, and hence can be solved by applying the simplex algorithm ([10]). Applying the simplex method instead to the dual linear program results in an increasing sequence of objective function values, each of which is a lower bound on the EMD. In contrast, all lower bounds presented in this report are independent of the algorithm used to compute the exact EMD.

3 Centroid-based Lower Bounds

The centroid $\overline{x}$ of the distribution $x = (X, w) \in \mathbf{D}^{d,n}$ is defined as

$$\overline{x} = \frac{\sum_{j=1}^{n} w_j x_j}{w_\Sigma}.$$

In section 3.1 we shall prove that the distance between the centroids of distributions is a lower bound on the EMD between distributions of equal weight. There is also, however, a centroid-based lower bound if the distributions are not equal weight. If $x = (X, w)$ is heavier than $y = (Y, u)$, then all of the weight in y is matched to part of the weight in x. The weight in x which is matched to y by an optimal flow is a sub-distribution x' of x. Formally, a *sub-distribution* $x' = (X', w')$ of $x = (X, w) \in \mathbf{D}^{d,n}$, denoted $x' \subset x$, is a distribution with $X' = X$ and $0 \leq w' \leq w$:

$$x' = \{ (x_1, w_1'), \ldots, (x_n, w_n') \} = (X, w') \in \mathbf{D}^{d,n}, \qquad 0 \leq w_j' \leq w_j \text{ for } j = 1, \ldots, n.$$

In words, the points of a sub-distribution x' are the same as the points of x and the weights of x' are bounded by the weights of x. One can visualize a sub-distribution $x' \subset x$ as the result of removing some of the dirt in the piles of dirt in x. The minimum distance between the centroid of y and the locus of the centroid of sub-distributions of x of total weight u_Σ is a lower bound on $\text{EMD}(x, y)$. Details are given in section 3.2.

3.1 Distributions of Equal Weight

Theorem 1 *Suppose $x = (X, w) \in \mathbf{D}^{d,m}$ and $y = (Y, u) \in \mathbf{D}^{d,n}$ are distributions of equal total weight $w_\Sigma = u_\Sigma$. Then*

$$\text{EMD}^{||\cdot||}(x, y) \geq ||\overline{x} - \overline{y}||.$$

Here the ground distance $||\cdot||$ is any L_p norm used to measure $d(x_i, y_j)$.

Proof The equal weight requirement implies that for any feasible flow $F = (f_{ij})$,

$$\sum_{i=1}^{m} f_{ij} = u_j \qquad \text{and} \tag{6}$$

$$\sum_{j=1}^{n} f_{ij} = w_i. \tag{7}$$

Then

$$\begin{aligned}
\left|\left|\sum_{i=1}^{m} w_i x_i - \sum_{j=1}^{n} u_j y_j\right|\right| &= \left|\left|\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} x_i - \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} y_j\right|\right| && ((6),(7)) \\
&= \left|\left|\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}(x_i - y_j)\right|\right| \\
&\leq \sum_{i=1}^{m}\sum_{j=1}^{n} ||f_{ij}(x_i - y_j)|| && (\Delta\text{-inequality}) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j|| && (f_{ij} \geq 0) \\
\left|\left|\sum_{i=1}^{m} w_i x_i - \sum_{j=1}^{n} u_j y_j\right|\right| &\leq \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||.
\end{aligned}$$

Dividing both sides of the last inequality by $w_\Sigma = u_\Sigma$ yields

$$||\overline{x} - \overline{y}|| \leq \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||}{w_\Sigma}$$

for any feasible flow F. Replacing F by a work minimizing flow gives the desired result. Note that this proof holds for every L_p distance/norm $||\cdot||$. ■

3.2 Distributions of Unequal Weight

Let $x = (X, w) \in \mathbf{D}^{d,m}$ and $y = (Y, u) \in \mathbf{D}^{d,n}$ be distributions with $w_\Sigma \geq u_\Sigma$. In any feasible flow $F = (f_{ij})$ from x to y, all of the weight u_j must be matched to weight in x

$$\sum_{i=1}^{m} f_{ij} = u_j,$$

and the total amount of matched weight is

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = u_\Sigma.$$

Let

$$x^F = \{\ (x_1, \sum_{j=1}^{n} f_{1j}), (x_2, \sum_{j=1}^{n} f_{2j}), \ldots, (x_m, \sum_{j=1}^{n} f_{mj})\ \} = (X, w^F).$$

Clearly, $w_\Sigma^F = u_\Sigma$. From the previous section we know that

$$\mathrm{EMD}(x^F, y) \geq \left|\left|\overline{x^F} - \overline{y}\right|\right|.$$

It follows that

$$\mathrm{EMD}(x^F, y) \geq \min_{F' \in \mathcal{F}(x,y)} \left|\left|\overline{x^{F'}} - \overline{y}\right|\right|, \tag{8}$$

where the minimum is taken over all feasible flows F' from x to y. Since (8) holds for every feasible flow F from x to y, we can replace F by a work minimizing flow F^* and obtain

$$\mathrm{EMD}(x,y) = \mathrm{EMD}(x^{F^*},y) \geq \min_{F' \in \mathcal{F}(x,y)} \left|\left|\overline{x^{F'}} - \overline{y}\right|\right|. \tag{9}$$

The minimum on the right-hand side of the inequality (9) can be re-stated as the minimum distance of the centroid of y to the centroid of any sub-distribution of x of total weight u_Σ:

$$\min_{F' \in \mathcal{F}(x,y)} \left|\left|\overline{x^{F'}} - \overline{y}\right|\right| = \min_{\substack{x' = (X,w') \subset x \\ w'_\Sigma = u_\Sigma}} \left|\left|\overline{x'} - \overline{y}\right|\right|. \tag{10}$$

Clearly, $x^{F'}$ is a sub-distribution of x with total weight u_Σ for every $F' \in \mathcal{F}(x,y)$. It remains to argue that any sub-distribution $x' \subset x$ with total weight u_Σ is $x^{F'}$ for some $F' \in \mathcal{F}(x,y)$. Since x' and y are equal-weight distributions, any one-to-one matching of the weights in x' and y defines a feasible flow between x' and y and, therefore, between x and y. Combining (9) and (10),

$$\mathrm{EMD}(x,y) \geq \min_{\substack{x' = (X,w') \subset x \\ w'_\Sigma = u_\Sigma}} \left|\left|\overline{x'} - \overline{y}\right|\right|. \tag{11}$$

In section 3.2.1 we show how this minimization problem can be formulated as the minimization of a quadratic function subject to linear constraints. However, solving this quadratic programming problem is likely to take more time than computing the EMD itself. In section 3.2.2 we show how to compute a bounding box for the locus of the centroid of any sub-distribution of x of total weight u_Σ. The minimum distance from the centroid of y to the bounding box is a lower bound of the EMD, although it is obviously not as tight as the lower bound in (11).

3.2.1 The Centroid Lower Bound

Given a distribution $x = (X,w) \in \mathbf{D}^{d,m}$, the locus of the centroid of sub-distributions of x of weight αw_Σ, $0 < \alpha \leq 1$, is

$$C^\alpha(x) = \left\{ \frac{\sum_{i=1}^m \tilde{w}_i x_i}{\tilde{w}_\Sigma} \ : \ 0 \leq \tilde{w}_i \leq w_i, \ 0 < \tilde{w}_\Sigma = \alpha w_\Sigma \right\}.$$

Let

$$v_i = \frac{\tilde{w}_i}{\tilde{w}_\Sigma} \qquad \text{and} \qquad \hat{w}_i = \frac{w_i}{\alpha w_\Sigma}.$$

Then

$$C^\alpha(x) = \left\{ \sum_{i=1}^m v_i x_i \ : \ 0 \leq v \leq \hat{w} = \frac{1}{\alpha}\frac{w}{w_\Sigma}, \ v_\Sigma = 1 \right\},$$

or, in terms of matrix multiplication,

$$C^\alpha(x) = \{ \, Xv \ : \ 0 \leq v \leq \hat{w} = \frac{1}{\alpha}\frac{w}{w_\Sigma}, \ 1^T v = 1 \, \}. \tag{12}$$

The symbol "1" is overloaded in the constraint $1^T v = 1$; on the left-hand side it is a vector of m ones, while on the right-hand side it is simply the integer one. It is easy to see from (12) that

$$C^{\alpha_1}(x) \supseteq C^{\alpha_2}(x) \quad \text{if } \alpha_1 \leq \alpha_2.$$

The locus $C^{\alpha}(x)$ is a convex polytope. The intersection of the halfspaces $v \geq 0$ and $v \leq \hat{w}$ is a convex polytope P_1. The intersection of P_1 with the hyperplane $1^T v = 1$ is another convex polytope P_2 of one dimension less. Finally, applying the linear map X to P_2 gives the convex polytope $C^{\alpha}(x)$. In [1], the authors characterize and provide algorithms to compute the locus $C_{L,H}(S)$ of the centroid of a set S of points with approximate weights, where weight w_i lies in a given interval $[l_i, h_i]$ and the total weight W is bounded as $L \leq W \leq H$. The locus $C^{\alpha}(x) = C_{1,1}(X)$ if $[l_i, h_i] = [0, \hat{w}_i]$.

Now suppose that $y = (Y, u) \in \mathbf{D}^{d,n}$ is a lighter distribution than x. In the previous section we argued that the EMD is bounded below by the minimum distance from $\overline{y}$ to a point in $C^{u_\Sigma/w_\Sigma}(x)$. We denote this minimum distance as CLOC(x, y) because it uses the locus of the centroid of sub-distributions of x of weight u_Σ. This lower bound can be computed by minimizing a quadratic objective function subject to linear constraints:

$$(\mathrm{CLOC}(x,y))^2 = \min_{v} \; ||Xv - \overline{y}||_2^2$$

subject to

$$\begin{aligned} v &\geq 0 \\ v &\leq \hat{w} = \frac{1}{u_\Sigma} w \\ 1^T v &= 1. \end{aligned}$$

The above minimization problem consists of m variables and $2m + 1$ linear constraints which are taken directly from (12).

3.2.2 The Centroid Bounding Box Lower Bound

As previously mentioned, the computation of the CLOC lower bound as described in the previous section is likely to require more time than an exact EMD computation. Yet the centroid locus $C^{\alpha}(x)$ can still be very useful in finding a fast to compute lower bound on the EMD. The idea is to precompute a bounding box $B^{\alpha}(x)$ for $C^{\alpha}(x)$ for a sample of α values, say $\alpha = 0.05k$ for $k = 1, \ldots, 20$. When given a lighter query distribution y at query time, the minimum distance from $\overline{y}$ to the bounding box $B^{\alpha_y}(x)$ is a lower bound on EMD(x, y), where α_y is the largest sample α value which does not exceed the total weight ratio u_Σ/w_Σ (the correctness of α_y follows from the containment property (14)). This lower bound computation will be very fast because the bounding boxes are precomputed and the query time computation of the minimum distance of the point $\overline{y}$ to the box $B^{\alpha_y}(x)$ is a constant time operation (it depends only on the dimension d, not the number of points in x or y).

If we write the matrix X in terms of its rows as

$$X = \begin{bmatrix} r_1^T \\ \vdots \\ r_d^T \end{bmatrix} \in \mathbf{R}^{d \times m},$$

then

$$Xv = \begin{bmatrix} r_1^T v \\ \vdots \\ r_d^T v \end{bmatrix} \in \mathbf{R}^{d}.$$

The computation of an axis-aligned bounding box for the centroid locus $C^\alpha(x)$ can be accomplished by solving the $2d$ linear programs

$$a_k = \min_v \, r_k^T v, \quad b_k = \max_v \, r_k^T v \qquad k = 1, \ldots, d$$

subject to

$$\begin{aligned} v &\geq 0 \\ v &\leq \hat{w} = \frac{1}{\alpha w_\Sigma} w \\ 1^T v &= 1. \end{aligned} \tag{13}$$

Each of these linear programs has m variables and $2m + 1$ constraints. The axis-aligned bounding box for the centroid locus $C^\alpha(x)$ is

$$B^\alpha(x) = \prod_{k=1}^{d} [a_k, b_k].$$

As with the true centroid loci $C^\alpha(x)$, we have a containment property for the bounding boxes $B^\alpha(x)$:

$$B^{\alpha_1}(x) \supseteq B^{\alpha_2}(x) \quad \text{if } \alpha_1 \leq \alpha_2. \tag{14}$$

This fact can be verified by observing that the constraints over which the minima a_k and maxima b_k are computed get weaker as α decreases (the only constraint involving α is (13)). Note also that the box $B^\alpha(x)$ includes its "interior" so that the lower bound $\mathrm{CBOX}(x, y)$ is zero if $\overline{y}$ lies "inside" $B^{\alpha_y}(x)$. Using the CBOX lower bound instead of the CLOC lower bound trades off computation speed for pruning power since the former is much faster to compute, but

$$\mathrm{EMD}(x, y) \geq \mathrm{CLOC}(x, y) \geq \mathrm{CBOX}(x, y).$$

Nevertheless, the pruning power of the CBOX lower bound will be high when the query distribution is well-separated from many of the database distributions (which implies that the centroids will also be well-separated).

4 Projection-based Lower Bounds

For v on the unit sphere S^{d-1} in $\mathbf{R}^d$, the projection $\mathrm{proj}_v(x)$ of the distribution $x = (X, w) \in \mathbf{R}^{d,m}$ along the direction v is defined as

$$\mathrm{proj}_v(x) = \{\ (v^T x_1, w_1), (v^T x_2, w_2), \ldots, (v^T x_m, w_m)\ \} = (v^T X, w) \in \mathbf{D}^{1,m}.$$

In words, the projection along v is obtained by using the lengths of the projections of the distribution points along v and leaving the corresponding weights unchanged. The following lemma shows that the EMD between projections is a lower bound on the EMD between the original distributions.

Lemma 1 *Let $v \in S^{d-1}$. Then*

$$\mathrm{EMD}(x, y) \geq \mathrm{EMD}(\mathrm{proj}_v(x), \mathrm{proj}_v(y)).$$

Proof This theorem follows easily from the definition of the EMD and the fact that

$$\begin{aligned} |v^T x_i - v^T y_j| &= |v^T(x_i - y_j)| \\ &= ||v||_2 \, ||x_i - y_j||_2 \, |\cos\theta_{v,(x_i-y_j)}| \\ &= ||x_i - y_j||_2 \, |\cos\theta_{v,(x_i-y_j)}| \\ |v^T x_i - v^T y_j| &\leq ||x_i - y_j||_2. \end{aligned}$$

■

The following theorem is an immediate consequence of Lemma 1.

Theorem 2 *Let* $V = \{v_1, \ldots, v_L\} \subset S^{d-1}$ *and*

$$\mathrm{PMAX}(V, x, y) = \max_{v \in V} \mathrm{EMD}(\mathrm{proj}_v(x), \mathrm{proj}_v(y))$$

Then

$$\mathrm{EMD}(x, y) \geq \mathrm{PMAX}(V, x, y).$$

For this lower bound to be of practical use, we must be able to compute it efficiently. In section 5, we present a straightforward, $\Theta(m+n)$ time algorithm to compute the EMD between equal-weight distributions on the line. In combination with Theorem 2, this algorithm provides the means to compute quickly a lower bound on the EMD between two equal-weight distributions.

One pruning strategy is to pick a set of random directions V along which to perform projections, and apply Theorem 2 to obtain a lower bound. The hope is that the differences between two distributions will be captured by looking along one of the directions in V. Another pruning strategy is to use the set of orthogonal axis directions for the set V. The following corollary is an immediate consequence of Theorem 2.

Corollary 1 *Let*

$$E = \{e_1, \ldots, e_d\} \subset S^{d-1}$$

be the set of axis directions, and let

$$\mathrm{PAMAX}(x, y) = \mathrm{PMAX}(E, x, y).$$

Then

$$\mathrm{EMD}(x, y) \geq \mathrm{PAMAX}(x, y).$$

Looking along the space axes is intuitively appealing when each axis measures a specific property. For example, suppose that distribution points are points in the CIE-Lab color space ([16]). If two images are very different in terms of the luminance values of pixels, then comparing the signature projections along the L-axis will reveal this difference and allow the system to avoid an exact EMD computation.

When the projection directions are the coordinate axes, we can prove a lower bound which involves the sum of the EMDs along axis directions.

Theorem 3 *If*

$$\mathrm{PASUM}(x, y) = \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \mathrm{EMD}(\mathrm{proj}_{e_k}(x), \mathrm{proj}_{e_k}(y)),$$

then

$$\mathrm{EMD}(x, y) \geq \mathrm{PASUM}(x, y).$$

Proof The proof uses the fact that

$$||a||_2 \geq \frac{1}{\sqrt{d}}||a||_1$$

for any vector $a \in \mathbf{R}^d$, a proof of which may be found in appendix I. It follows that

$$\begin{aligned}
\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||_2 &\geq \frac{1}{\sqrt{d}}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||_1 \\
&= \frac{1}{\sqrt{d}}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\sum_{k=1}^{d}\left|x_i^{(k)} - y_j^{(k)}\right| \\
&= \frac{1}{\sqrt{d}}\sum_{k=1}^{d}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\left|x_i^{(k)} - y_j^{(k)}\right| \\
\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||_2 &\geq \frac{1}{\sqrt{d}}\sum_{k=1}^{d}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\left|x_i^{(k)} - y_j^{(k)}\right|,
\end{aligned}$$

where the superscript (k) denotes the kth component of a vector. Therefore,

$$\begin{aligned}
\min_{F\in\mathcal{F}(x,y)}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||_2 &\geq \min_{F\in\mathcal{F}(x,y)}\frac{1}{\sqrt{d}}\sum_{k=1}^{d}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\left|x_i^{(k)} - y_j^{(k)}\right| \\
&\geq \frac{1}{\sqrt{d}}\sum_{k=1}^{d}\min_{F\in\mathcal{F}(x,y)}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\left|x_i^{(k)} - y_j^{(k)}\right| \\
&= \frac{1}{\sqrt{d}}\sum_{k=1}^{d}(\min(w_\Sigma, u_\Sigma) \times \text{EMD}(\text{proj}_{e_k}(x), \text{proj}_{e_k}(y))) \\
&= \frac{1}{\sqrt{d}}\min(w_\Sigma, u_\Sigma)\sum_{k=1}^{d}\text{EMD}(\text{proj}_{e_k}(x), \text{proj}_{e_k}(y)) \\
\min_{F\in\mathcal{F}(x,y)}\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}||x_i - y_j||_2 &\geq \frac{1}{\sqrt{d}}\min(w_\Sigma, u_\Sigma)\sum_{k=1}^{d}\text{EMD}(\text{proj}_{e_k}(x), \text{proj}_{e_k}(y)).
\end{aligned}$$

Dividing both sides of the last inequality by $\min(w_\Sigma, u_\Sigma)$ gives the desired result. ■

Note that PASUM(x, y) may be rewritten as

$$\text{PASUM}(x, y) = \sqrt{d}\left(\frac{\sum_{k=1}^{d}\text{EMD}(\text{proj}_{e_k}(x), \text{proj}_{e_k}(y))}{d}\right).$$

This alternate expression makes it clear that PASUM(x, y) is a better lower bound than PAMAX(x, y) iff the square root of the dimension times the average axis projection distance is greater than the maximum axis projection distance.

5 The EMD in One Dimension

Let $x = (X, w) \in \mathbf{D}^{1,m}$ and $y = (Y, u) \in \mathbf{D}^{1,n}$ be distributions on the real line. Assume the points in x and y are sorted by position:

$$x_1 < x_2 < \cdots < x_m \qquad \text{and} \qquad y_1 < y_2 < \cdots < y_n.$$

Define the *cumulative distribution function* (CDF) of x as

$$W(t) = \begin{cases} 0 & \text{if } t \in (-\infty, x_1) \\ \sum_{i=1}^{k} w_i & \text{if } t \in [x_k, x_{k+1}),\ \ 1 \le k \le m-1 \\ w_\Sigma = \sum_{i=1}^{m} w_i & \text{if } t \in [x_m, \infty). \end{cases}$$

Similarly, the CDF of y is

$$U(t) = \begin{cases} 0 & \text{if } t \in (-\infty, y_1) \\ \sum_{j=1}^{l} u_j & \text{if } t \in [y_l, y_{l+1}),\ \ 1 \le l \le n-1 \\ u_\Sigma = \sum_{j=1}^{n} u_j & \text{if } t \in [y_n, \infty). \end{cases}$$

If x and y are equal weight, then the work to transform one distribution into the other is the area between the graphs of the CDFs of x and y. See figure 1. We will now prove

Theorem 4 *If $x = (X, w) \in \mathbf{D}^{1,m}$ and $y = (Y, u) \in \mathbf{D}^{1,n}$ have equal weight $w_\Sigma = u_\Sigma$, then*

$$\mathrm{EMD}(x, y) = \frac{\int_{-\infty}^{\infty} |W(t) - U(t)|\, dt}{w_\Sigma}.$$

Proof Let

$$r_1 < r_2 < \cdots < r_{m+n}$$

be the sorted list of breakpoints $x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n$. Note that $W(t)$ and $U(t)$ are constant over the interval $t \in [r_k, r_{k+1})$ for $k = 1, \ldots, m+n-1$, $W(t) = U(t) \equiv 0$ for $t \in (-\infty, r_1)$, and $W(t) = U(t) \equiv w_\Sigma = u_\Sigma$ for $t \in [r_{m+n}, \infty)$. Therefore the integral of the absolute difference of the CDFs may be written as the finite summation

$$\int_{-\infty}^{\infty} |W(t) - U(t)|\, dt = \sum_{k=1}^{m+n-1} (r_{k+1} - r_k)\, |W(r_k) - U(r_k)|.$$

We claim that there is exactly one feasible flow F that can morph x into y. Consider the interval (r_k, r_{k+1}). At any position t in this interval, the absolute difference $|W(t) - U(t)|$ is equal to $|W(r_k) - U(r_k)|$. Suppose that $W(r_k) > U(r_k)$. Then in any feasible flow from x to y, exactly $W(r_k) - U(r_k)$ weight from x must be moved from r_k to r_{k+1}. If less than this amount is moved, then there will be less x weight than y weight in $[r_{k+1}, \infty)$ after the flow is complete. If more than this amount is moved, then there will be more x weight than y weight in $[r_{k+1}, \infty)$ after the flow is complete. Moving weight from r_{k+1} to r_k would only increase the surplus of x weight in $(-\infty, r_k]$. See figure 2(a). Similar logic shows that if $U(r_k) > W(r_k)$, then exactly $U(r_k) - W(r_k)$ weight from x must be moved from r_{k+1} to r_k. This case is illustrated in figure 2(b). In either case, the amount of work E_k done in moving weight from x over the interval (r_k, r_{k+1}) is

$$E_k = (r_{k+1} - r_k)\, |W(r_k) - U(r_k)|.$$

The total work E performed in the unique feasible flow from x to y is

$$E = \sum_{k=1}^{m+n-1} E_k.$$

It follows that

$$\mathrm{EMD}(x, y) = \frac{E}{w_\Sigma},$$

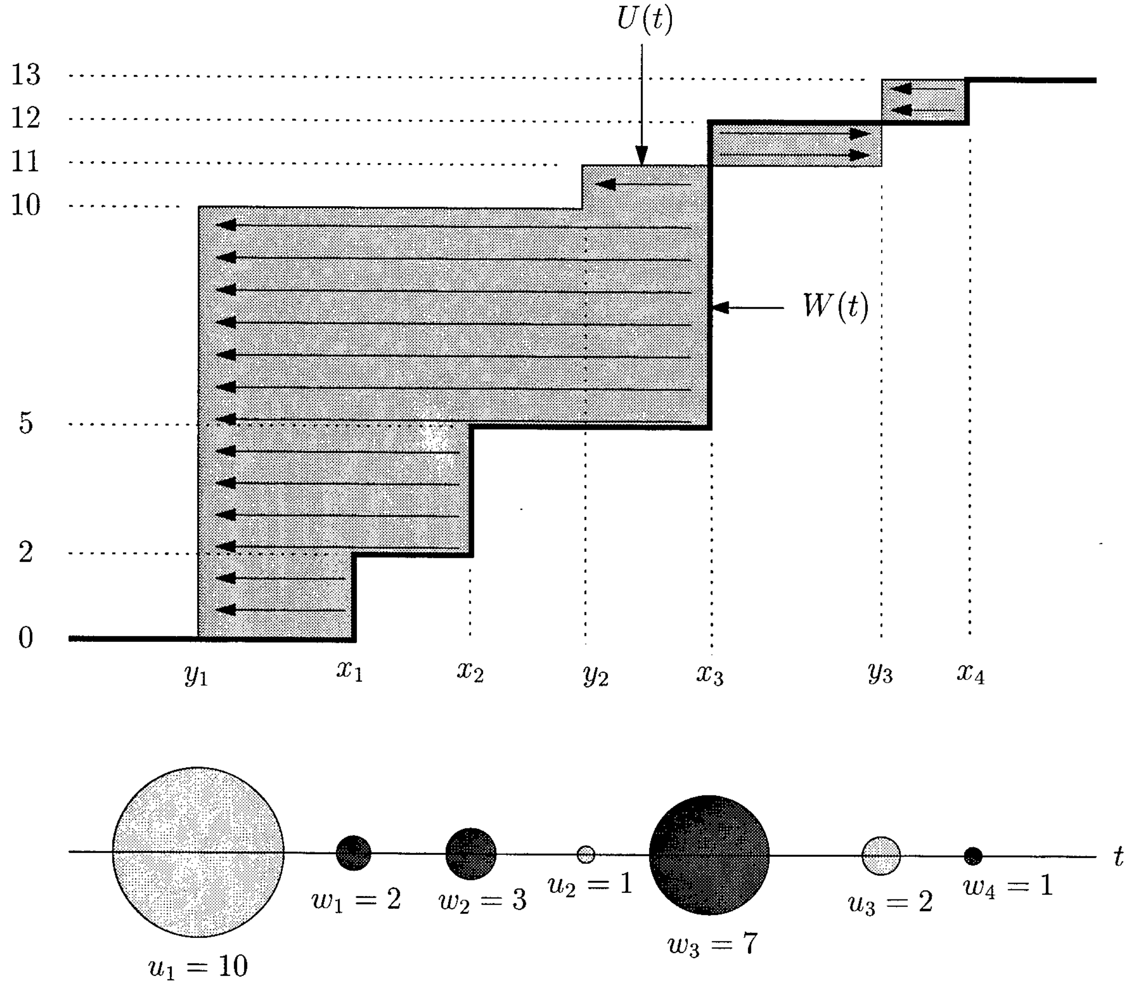


Figure 1: The cumulative distribution functions (CDFs) for the equal-weight line distributions x and y are $W(t)$ and $U(t)$, repsectively. The work to transform x into y is equal to the area between the two CDFs. The unique transforming flow is shown with directed lines from x weight to the matching y weight. The EMD between x and y is obtained by dividing the work by the total weight of the distributions ($w_\Sigma = u_\Sigma = 13$ in the picture).

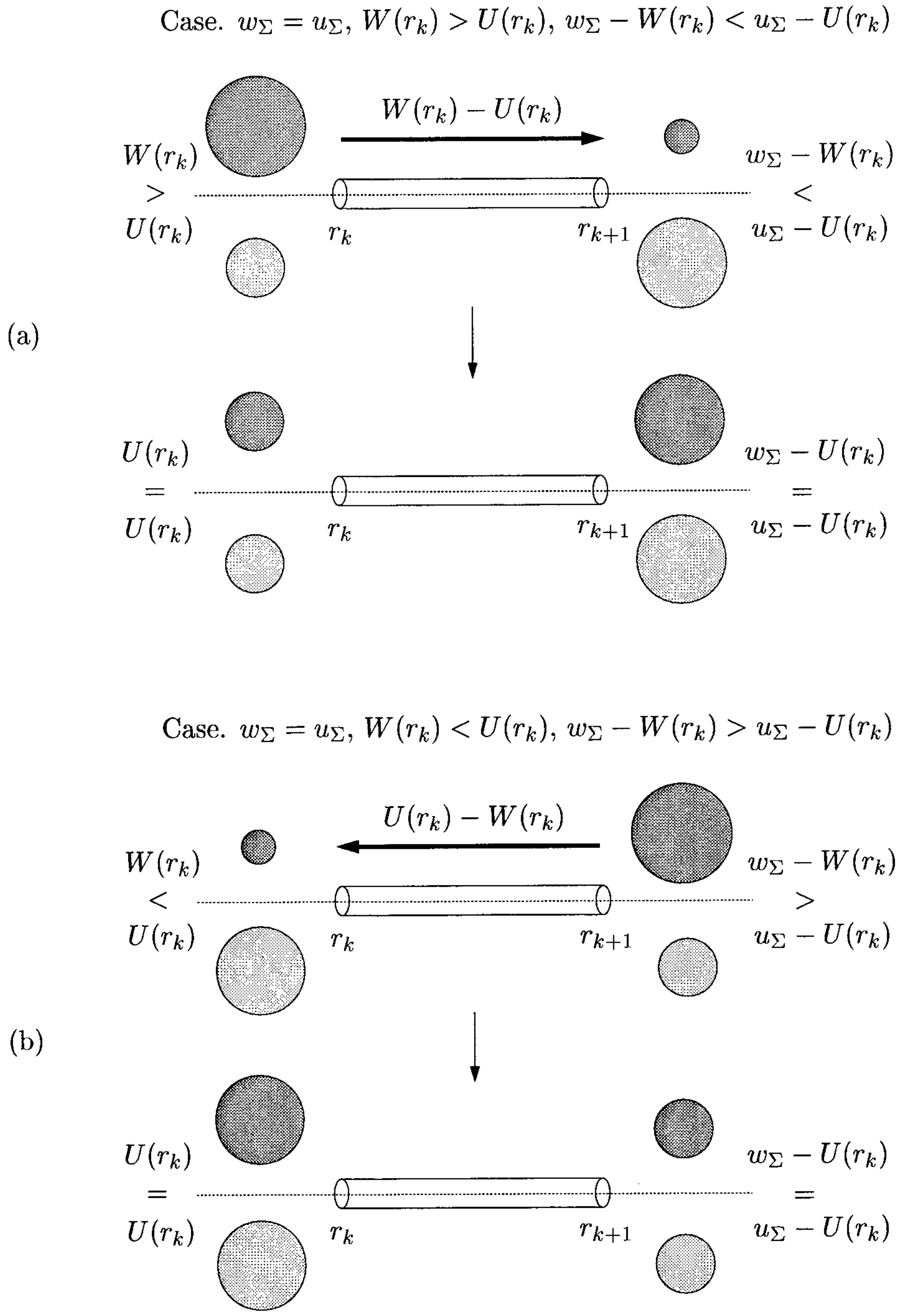


Figure 2: The unique feasible flow between equal-weight distributions $x = (X, w)$ and $y = (Y, u)$ on the line. Here $r_1 < \cdots < r_{m+n}$ is the position-sorted list of points in x and y, and $W(t)$ and $U(t)$ are the CDFs for x and y, respectively. (a) $W(r_k) > U(r_k)$, $w_\Sigma - W(r_k) < u_\Sigma - U(r_k)$. In this case, a flow from x to y is feasible only if exactly $W(r_k) - U(r_k)$ of x weight travels from r_k to r_{k+1} during the flow. (b) $W(r_k) < U(r_k)$, $w_\Sigma - W(r_k) > u_\Sigma - U(r_k)$. In this case, a flow from x to y is feasible only if exactly $U(r_k) - W(r_k)$ of x weight travels from r_{k+1} to r_k during the flow.

and this completes the proof. ■

When the weights of the distributions are unequal, there is no longer a unique feasible flow. However, arguments similar to those used above can be used to compute a lower bound on any feasible flow. Once again consider the interval (r_k, r_{k+1}), and WLOG assume $w_\Sigma > u_\Sigma$ and that x weight is moved to match all the y weight. When there is more x weight than y weight in both $(-\infty, r_k]$ and $[r_{k+1}, \infty)$, then there will be feasible flows in which no x weight travels through (r_k, r_{k+1}). If there is more x weight than y weight in $(-\infty, r_k]$, but less x weight than y weight in $[r_{k+1}, \infty)$, then $(u_\Sigma - U(r_k)) - (w_\Sigma - W(r_k))$ of the x weight must be moved from r_k to r_{k+1} in order to cover the y weight in $[r_{k+1}, \infty)$. See figure 3(a). If there is less x weight than y weight in $(-\infty, r_k]$, but more x weight than y weight in $[r_{k+1}, \infty)$, then $U(r_k) - W(r_k)$ of the x weight must be moved from r_{k+1} to r_k in order to cover the y weight in $(-\infty, r_k]$. This case is illustrated in figure 3(b). Under the assumption that $w_\Sigma > u_\Sigma$, it *cannot* be the case that there is less x weight than y weight in both $(-\infty, r_k]$ and $[r_{k+1}, \infty)$.

Pseudocode for the lower bound described in the previous paragraph is given below. The routine is named FSBL because the lower bound follows simply from flow feasibility (FeaSiBiLity) conditions.

```
function FSBL($x, y$) := /* assumes $d = 1$, $w_\Sigma \geq u_\Sigma$ */
   work = 0
   $r_1 = \min(x_1, y_1)$
   for $k = 1$ to $m + n - 1$
      $r_{k+1}$ = smallest point in $x$ or $y$ that is greater than $r_k$
      if $u_\Sigma - U(r_k) > w_\Sigma - W(r_k)$ then
         work += $((u_\Sigma - U(r_k)) - (w_\Sigma - W(r_k))) \times (r_{k+1} - r_k)$
      elseif $U(r_k) > W(r_k)$ then
         work += $(U(r_k) - W(r_k)) \times (r_{k+1} - r_k)$
      end if
   end for
   return (work / $u_\Sigma$)
end function
```

We have argued that

Theorem 5 *If x and y are distributions on the line, then*

$$\mathrm{EMD}(x, y) \geq \mathrm{FSBL}(x, y).$$

If $w_\Sigma = u_\Sigma$, then $(u_\Sigma - U(r_k) > w_\Sigma - W(r_k)) \equiv (W(r_k) > U(r_k))$, $(u_\Sigma - U(r_k)) - (w_\Sigma - W(r_k)) = W(r_k) - U(r_k)$, and the routine computes the exact value $\mathrm{EMD}(x, y)$.

Theorem 6 *If x and y are two equal-weight distributions on the line, then*

$$\mathrm{EMD}(x, y) = \mathrm{FSBL}(x, y).$$

Assuming that the points in $x \in \mathbf{D}^{1,m}$ and $y \in \mathbf{D}^{1,n}$ are in sorted order, the routine runs in linear time $\Theta(m+n)$. The combined sorted list $r_1, \ldots, r_{m+n}$ of points in x and y is discovered by walking along the two sorted lists of points. At any time during the algorithm, there is a pointer to the

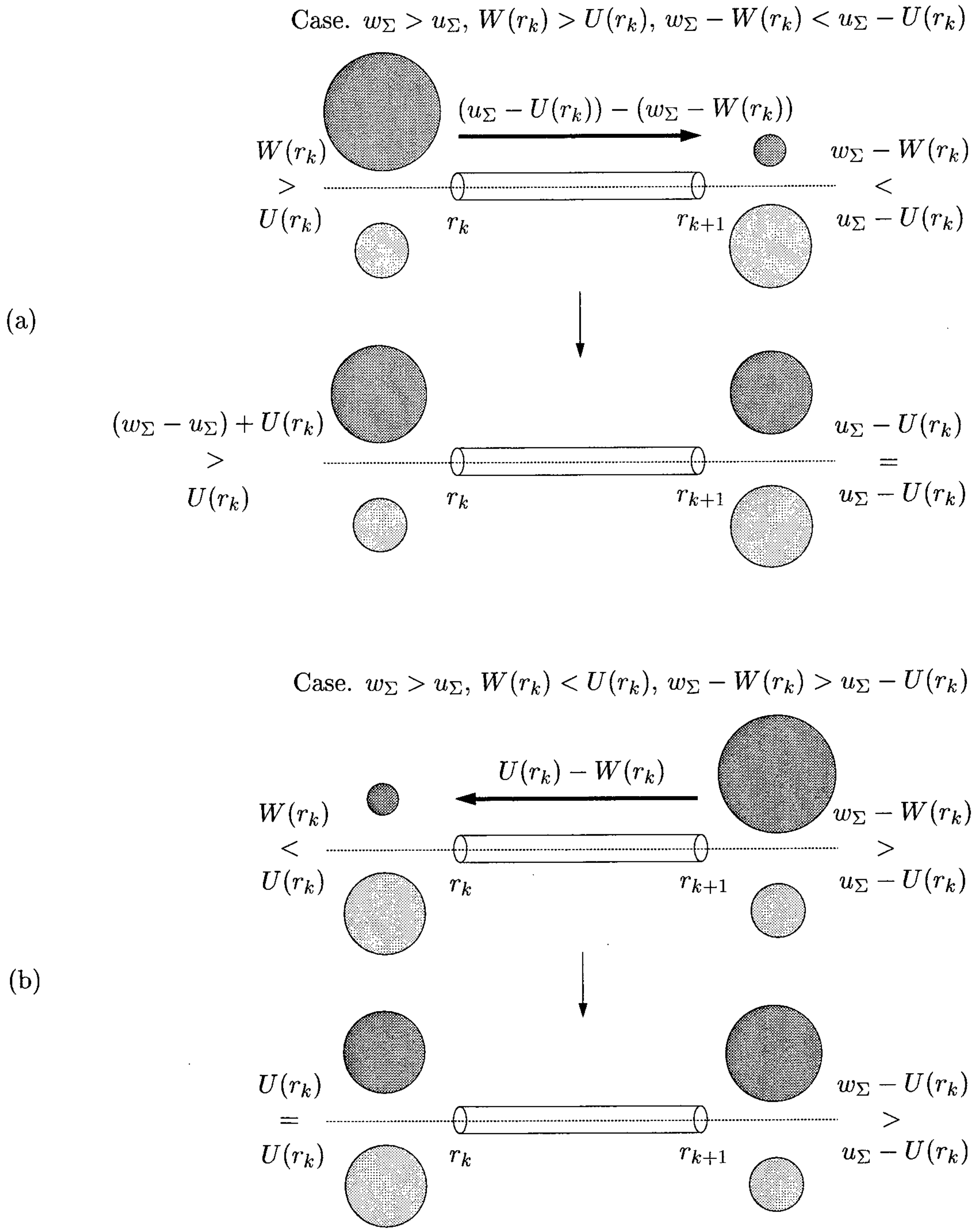


Figure 3: Necessary conditions for a feasible flow between unequal-weight distributions $x = (X, w)$ and $y = (Y, u)$ on the line, where $w_\Sigma > u_\Sigma$. All y weight must be covered by x weight. (a) $W(r_k) > U(r_k)$, $w_\Sigma - W(r_k) < u_\Sigma - U(r_k)$. In this case, a necessary condition to have a feasible flow from x to y is that at least $(w_\Sigma - W(r_k)) - (u_\Sigma - U(r_k))$ of x weight travels from r_k to r_{k+1} during the flow. (b) $W(r_k) < U(r_k)$, $w_\Sigma - W(r_k) > u_\Sigma - U(r_k)$. In this case, a necessary condition to have a feasible flow from x to y is that at least $U(r_k) - W(r_k)$ of x weight travels from r_{k+1} to r_k during the flow.

next x and next y value to be considered. The value r_{k+1} then follows in constant time from the value of r_k.

The FSBL lower bound may be substituted for the EMD function in the PMAX, PAMAX, and PASUM lower bounds to obtain efficient to compute, projection-based lower bounds

$$\begin{aligned}
\mathrm{PMAX}_{\mathrm{FSBL}}(V,x,y) &= \max_{v\in V} \mathrm{FSBL}(\mathrm{proj}_v(x), \mathrm{proj}_v(y)) \\
&= \mathrm{PMAX}(V,x,y) \qquad \text{when } w_\Sigma = u_\Sigma
\end{aligned}$$

$$\begin{aligned}
\mathrm{PAMAX}_{\mathrm{FSBL}}(x,y) &= \max_{k=1,\ldots,d} \mathrm{FSBL}(\mathrm{proj}_{e_k}(x), \mathrm{proj}_{e_k}(y)) \\
&= \mathrm{PAMAX}(x,y) \qquad \text{when } w_\Sigma = u_\Sigma
\end{aligned}$$

$$\begin{aligned}
\mathrm{PASUM}_{\mathrm{FSBL}}(x,y) &= \frac{1}{\sqrt{d}} \sum_{k=1}^{d} \mathrm{FSBL}(\mathrm{proj}_{e_k}(x), \mathrm{proj}_{e_k}(y)) \\
&= \mathrm{PASUM}(x,y) \qquad \text{when } w_\Sigma = u_\Sigma
\end{aligned}$$

in which x and y are not necessarily equal weight. The second equality in each of the three pairs of equalities follows directly from Theorem 6 and the definitions of PMAX(V,x,y), PAMAX(x,y), and PASUM(x,y).

6 Experiments in Color-based Retrieval

In this section, we show some results of using the lower bounds CBOX, $\mathrm{PMAX}_{\mathrm{FSBL}}$, $\mathrm{PAMAX}_{\mathrm{FSBL}}$, and $\mathrm{PASUM}_{\mathrm{FSBL}}$ in the color-based retrieval system described in [11]. This system summarizes an image by a distribution of dominant colors in the CIE-Lab color space, where the weight of a dominant color is equal to the fraction of image pixels which are classified as that color. The input to the system is a query and a number K of nearest images to return. The system computes the EMD between the query distribution and each of the database distributions. If the query is a full image (e.g. an image in the database), then the query distribution and all the database distributions will have total weight equal to one. In this query-by-example setting, the system first checks the distance between distribution centroids before performing an exact EMD computation. If the centroid distance is larger than the Kth largest distance seen before the current comparison, then the system does not compute the EMD and simply considers the next database image. A K-nearest neighbor database image to the query cannot be missed by this algorithm because the centroid distance is a lower bound on the EMD between equal-weight distributions. When the query is a partial query (such as "give me all the images with at least 20% sky blue"), an exact EMD computation is performed between the query and every database image.

To use the CBOX lower bound for partial queries, some additional preprocessing is needed. At database entry time, the distribution $x = (X, w)$ of an image is computed and stored, as well as the centroid bounding boxes $B^{\alpha}(x)$ for $\alpha = 0.05k$, $k = 1, \ldots, 20$. Given a query distribution $y = (Y, u)$ of weight $u_\Sigma \leq w_\Sigma$, let α_y denote the largest sample α value which does not exceed the total weight ratio u_Σ / w_Σ. The system computes the distance between $\overline{y}$ and the nearest point in $B^{\alpha_y}(x)$. This is the CBOX lower bound. To use the $\mathrm{PMAX}_{\mathrm{FSBL}}$ lower bound, a set V of L (specified later) random projection directions and the L position-sorted projections of each database distribution along the directions in V are computed and stored at database load time. At query time, the query distribution is also projected along the directions in V. To use the $\mathrm{PAMAX}_{\mathrm{FSBL}}$ and $\mathrm{PASUM}_{\mathrm{FSBL}}$

lower bounds, the d position-sorted projections of each database distribution along the space axes are computed and stored at database entry time. At query time, the same axis projections are performed on the query distribution.

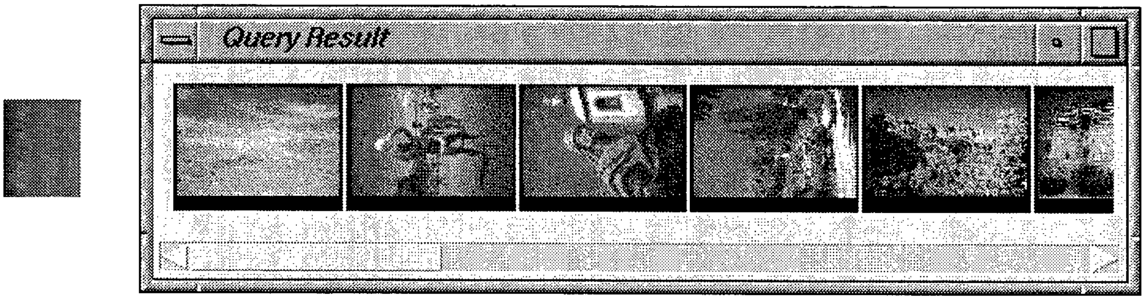
There are many factors that affect the performance of our lower bounds. The most obvious is the database itself. Here, we use a Corel database of 20000 color images which is dominated by outdoor scenes. The order in which the images are compared to the query is also important. If the most similar images to a query are processed first, then the Kth smallest distance seen will be relatively small when the dissimilar images are processed, and relatively weak lower bounds can prune these dissimilar images. Of course, the purpose of the query is to discover the similar images. Nonetheless, a random order of comparison may help ensure good performance over a wide range of queries. Moreover, if a certain type of query is more likely than others, say, for example, queries with large amounts of blue and green (to retrieve outdoor images containing sky and grass), then it would be wise to pre-determine a good comparison order to use for such queries. In the results that follow, the comparison order is the same for all queries, and the order is *not* specialized for any particular type of query.

The number K of nearest images to return is yet another factor. For a fixed comparison order and query, the number of exact EMD calculations pruned is inversely related to the size of K. This is because the Kth smallest distance after comparing a certain number images, against which a lower bound is compared, is an increasing function of K. In all the upcoming experiments, the number of nearest images returned is fixed at $K = 20$. In terms of the actual lower bounds, a system may be able to achieve better query times by using more than one bound. For example, a system might apply the CBOX lower bound first, followed by the more expensive $\text{PASUM}_{\text{FSBL}}$ bound if CBOX fails, followed by an even more expensive exact EMD computation if $\text{PASUM}_{\text{FSBL}}$ also fails. The hope is that the lower bound hierarchy of CBOX, $\text{PASUM}_{\text{FSBL}}$, and EMD speeds up query times in much the same way that the memory hierarchy of primary cache, secondary cache, and main memory speeds up memory accesses. Our experiments, however, apply one lower bound per query. For the $\text{PMAX}_{\text{FSBL}}$ lower bound, the number L of random directions must be specified. This parameter trades off between pruning power and computation speed. The more directions, the greater the pruning power, but the slower the computation. In our work, we use the heuristic $L = 2d$ (without quantifiable justification), where d is the dimension of the underlying point space (so $L = 6$ in the color-based system).

All experiments were conducted on an SGI Indigo^2 with a 250 MHz processor, and query times are reported in seconds (s). The exact EMD is computed via an efficient solution to the transportation problem based on the work [6]. The color signature of a typical database image has eight to twelve points. The time for an EMD calculation between two such images varies roughly between half a millisecond and one millisecond (ms). The EMD computation time increases with the number of points in the distributions, so EMD computations involving a partial query distribution with only a few points are, in general, faster than EMD computations between two database images. The time for an EMD computation between a database image and a partial query with three or fewer points is typically about 0.25ms.

We begin our experiments with a few very simple queries. Each of these queries consists of a distribution with exactly one color point in CIE-Lab space. The results of the three queries

(a)



(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	2.210
CBOX	19675	0.193
$\text{PMAX}_{\text{FSBL}}$	19715	0.718
$\text{PAMAX}_{\text{FSBL}}$	19622	0.441
$\text{PASUM}_{\text{FSBL}}$	18969	0.536

Figure 4: Query C.1.1 – 20% blue. (a) query results. (b) query statistics.

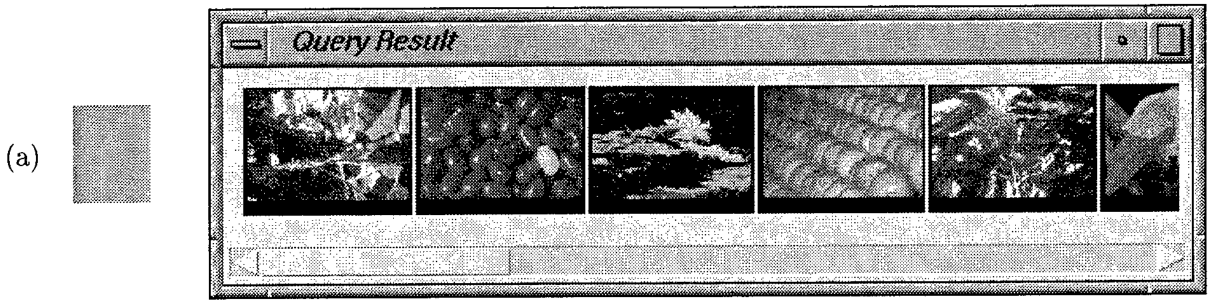
C.1.1 at least 20% (sky) blue ▨ ,

C.1.2 at least 40% green ▨ , and

C.1.3 at least 60% red ▨

are shown in figure 4, figure 5, and figure 6, respectively. In these examples, all the lower bounds result in query times which are less than the brute force query time, and avoid a large fraction of exact EMD computations. The CBOX and $\text{PASUM}_{\text{FSBL}}$ bounds gave the best results on these three queries.

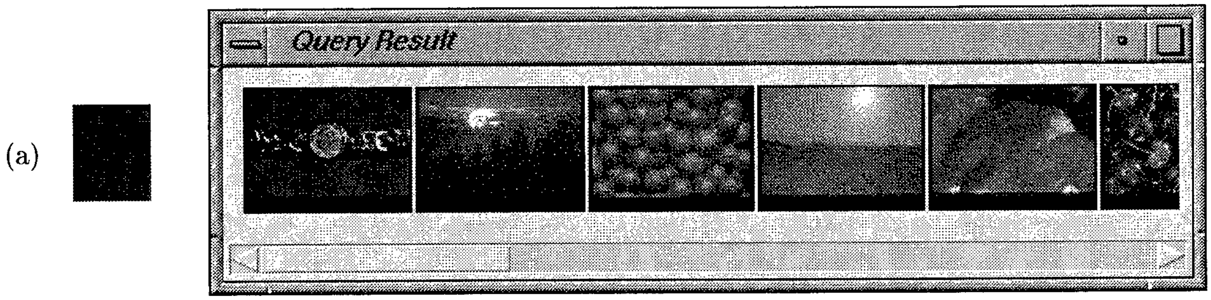
The next set of examples consists of randomly generated partial queries. The results for the five queries



(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	3.043
CBOX	19634	0.233
$\text{PMAX}_{\text{FSBL}}$	10172	2.552
$\text{PAMAX}_{\text{FSBL}}$	16222	1.124
$\text{PASUM}_{\text{FSBL}}$	18424	0.754

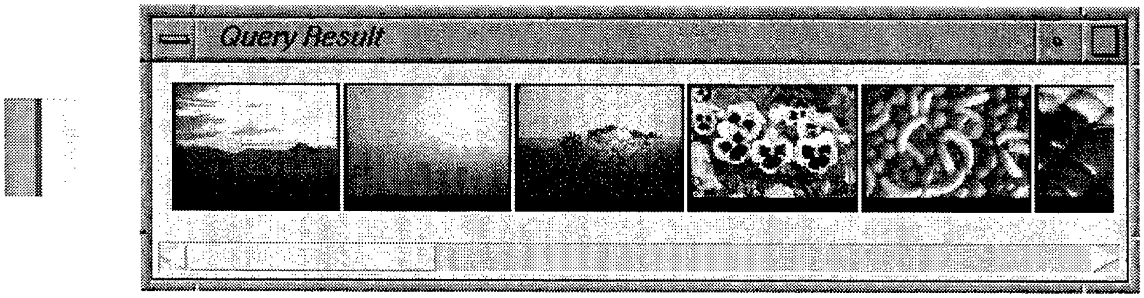
Figure 5: Query C.1.2 – 40% green. (a) query results. (b) query statistics.



(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	2.920
CBOX	19621	0.240
$\text{PMAX}_{\text{FSBL}}$	15903	1.505
$\text{PAMAX}_{\text{FSBL}}$	17125	0.871
$\text{PASUM}_{\text{FSBL}}$	18182	0.785

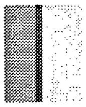
Figure 6: Query C.1.3 – 60% red. (a) query results. (b) query statistics.

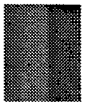
(a) 

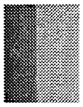
(b)

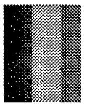
Lower Bound	# Pruned	Query Time (s)
NONE	0	4.240
CBOX	18704	0.496
$\text{PMAX}_{\text{FSBL}}$	17989	1.323
$\text{PAMAX}_{\text{FSBL}}$	17784	1.035
$\text{PASUM}_{\text{FSBL}}$	18418	0.832

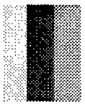
Figure 7: Query C.2.1 – 13.5% green, 3.4%red, 17.8% yellow. (a) query results. (b) query statistics.

C.2.1 13.5% green, 3.4%red, 17.8% yellow  ,

C.2.2 26.0% blue, 19.7% violet  ,

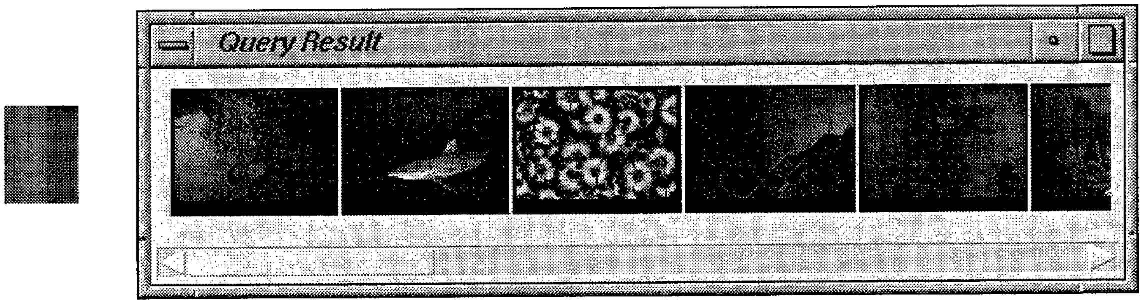
C.2.3 16.8% blue, 22.2% green, 1.8% yellow  ,

C.2.4 22.8% red, 24.2% green, 17.3% blue  , and

C.2.5 13.2% yellow, 15.3% violet, 15.3% green 

are shown in figure 7 through figure 11, respectively. The CBOX lower bound gives the best results for queries C.2.1 and C.2.2, but its performance drops by an order of magnitude for C.2.3, and it is completely ineffective for C.2.4 and C.2.5. Indeed, the CBOX lower bound pruned only 1 of 20000 database images for query C.2.5. The CBOX behavior can be explained in part by the locations of centroids of the query distributions and the database distributions. See figure 12. Roughly speaking, the effectiveness of the CBOX bound is directly related to the amount of separation between the database distributions and the query distribution, with larger separation implying a more effective bound. The query C.2.1 consists almost entirely of green and yellow. As one

(a)

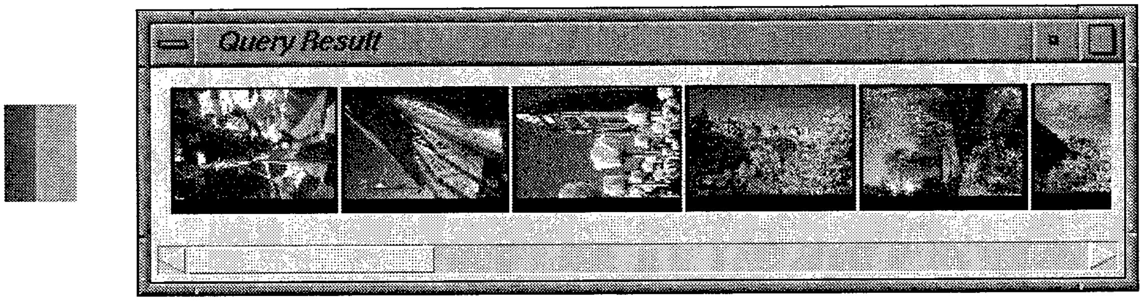


(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	3.812
CBOX	18631	0.453
$\text{PMAX}_{\text{FSBL}}$	16472	1.452
$\text{PAMAX}_{\text{FSBL}}$	17032	1.010
$\text{PASUM}_{\text{FSBL}}$	17465	1.037

Figure 8: Query C.2.2 – 26.0% blue, 19.7% violet, (a) query results. (b) query statistics.

(a)

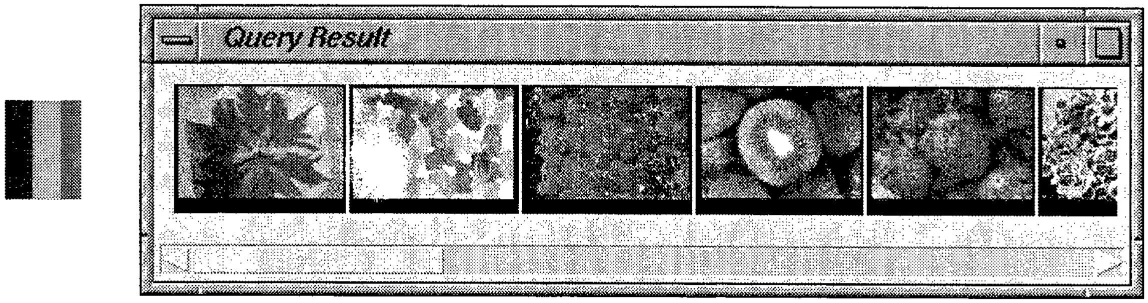


(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	4.073
CBOX	1631	3.999
$\text{PMAX}_{\text{FSBL}}$	10550	3.235
$\text{PAMAX}_{\text{FSBL}}$	11690	2.648
$\text{PASUM}_{\text{FSBL}}$	15386	1.612

Figure 9: Query C.2.3 – 16.8% blue, 22.2% green, 1.8% yellow. (a) query results. (b) query statistics.

(a)

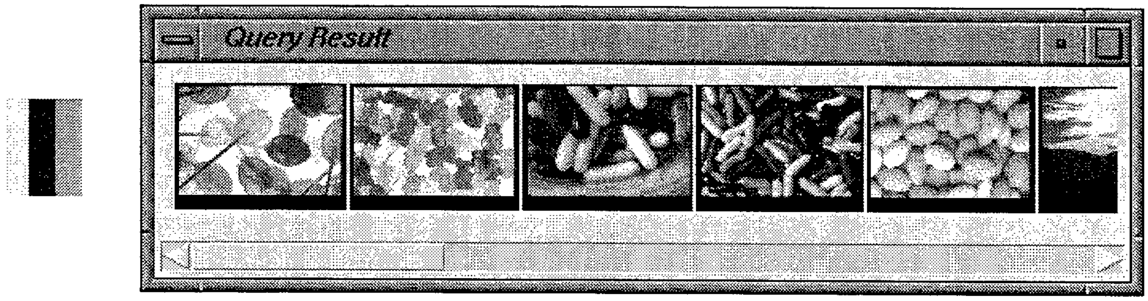


(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	3.969
CBOX	26	4.158
$\text{PMAX}_{\text{FSBL}}$	3606	4.342
$\text{PAMAX}_{\text{FSBL}}$	3399	4.010
$\text{PASUM}_{\text{FSBL}}$	12922	2.324

Figure 10: Query C.2.4 – 22.8% red, 24.2% green, 17.3% blue. (a) query results. (b) query statistics.

(a)



(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	3.375
CBOX	1	3.560
$\text{PMAX}_{\text{FSBL}}$	9608	2.924
$\text{PAMAX}_{\text{FSBL}}$	10716	2.381
$\text{PASUM}_{\text{FSBL}}$	15562	1.492

Figure 11: Query C.2.5 – 13.2% yellow, 15.3% violet, 15.3% green. (a) query results. (b) query statistics.

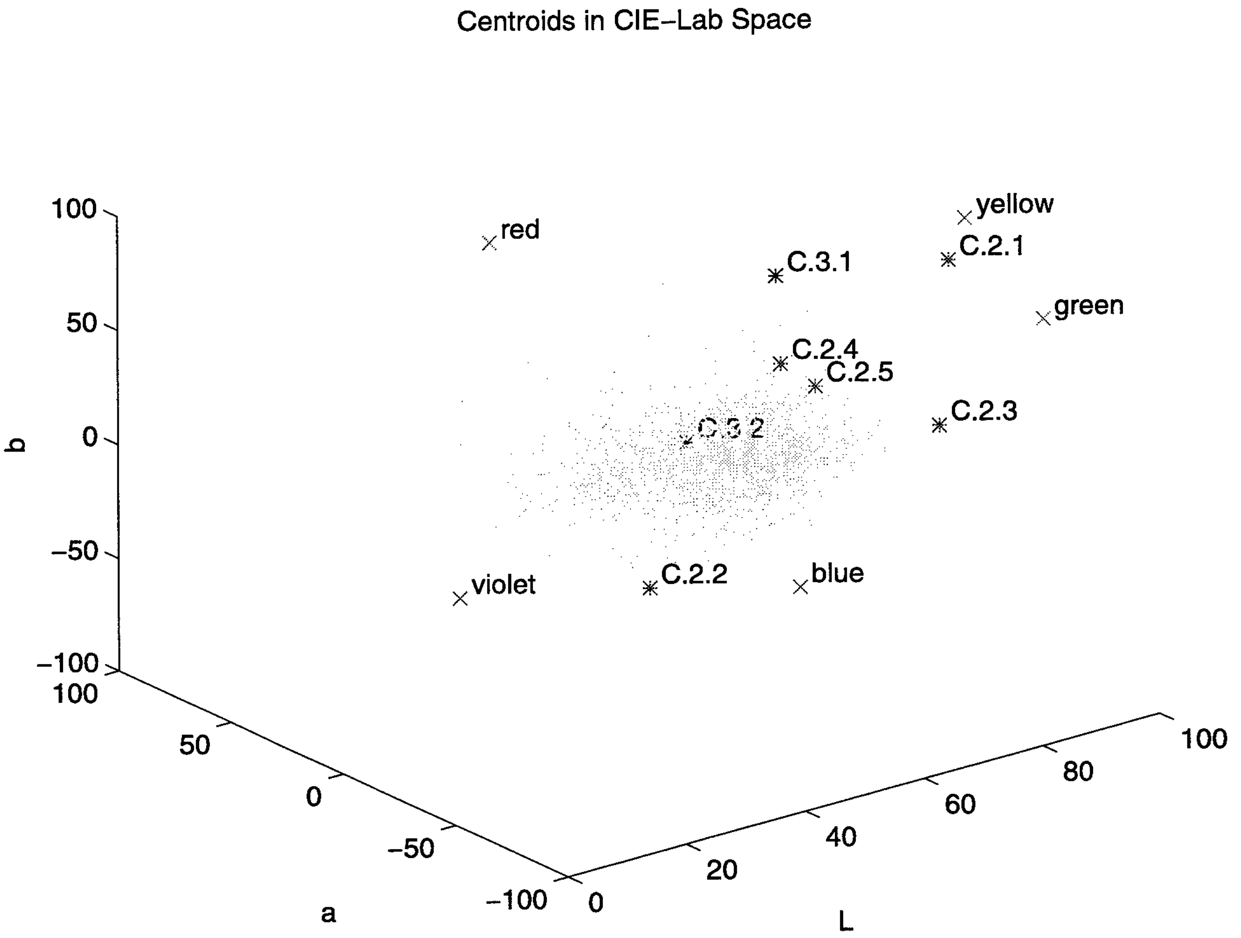


Figure 12: The centroids of the color signature distributions of a random subset of 5000 images in the Corel database are plotted as dots, and the centroids for the queries C.2.* and C.3.* are plotted as stars. The locations of blue (C.1.1), green (C.1.2), red (C.1.3), yellow, and violet are plotted as x's.

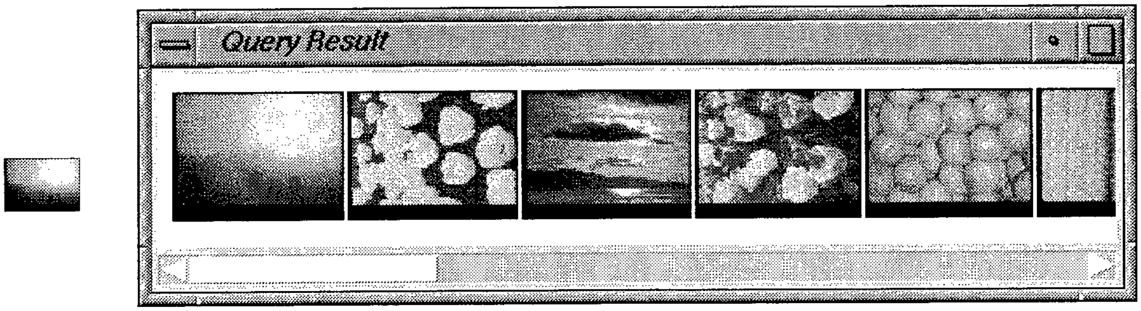
can see from figure 12, the centroid of C.2.1 is very isolated from the database centroids. The approximately equal amounts red, green, and blue in query C.2.4 result in a centroid which is close to a large number of database centroids. The same statement holds for query C.2.5 which has green and yellow in one corner of the CIE-Lab space, and violet at the opposite corner.

The distances of the centroids for C.2.2 and C.2.3 to the database centroids are (i) about the same, and (ii) are smaller than the distance for C.2.1 and larger than the distances for C.2.4 and C.2.5. Observation (ii) helps explain why the performance of CBOX on C.2.2 and C.2.3 is worse than the performance on C.2.1, but better than the performance on C.2.4 and C.2.5. Observation (i) might lead one to believe that the CBOX performance should be about the same on C.2.2 and C.2.3. The statistics, however, show that this is not the case. To understand why, we must remember that the queries are partial queries. The relevant quantity is not the centroid of a database distribution, but rather the locus of the centroid of all sub-distributions with weight equal to the weight of the query. Consider images with significant amounts of blue and green, and other colors which are distant from blue and green (such as red). The other colors will help move the distribution centroid away from blue and green. However, a sub-distribution of such an image which contains only blue and green components will have a centroid which is close to blue and green, and hence close to the centroid of C.2.3. The distance between the query centroid and this image centroid may be large, but the CBOX lower bound will be small (and, hence, weak). From figure 12 and the results of C.2.2 and C.2.3, one can infer that there are many more images that contain blue, green, and significant amounts of distant colors from blue and green than there are images that contain blue, violet, and significant amounts of distant colors from blue and violet. The centroid is a measure of the (weighted) average color in a distribution, and the average is not an accurate representative of a distribution with high variance (i.e. with colors that span a large portion of the color space).

The projection-based lower bounds $\mathrm{PMAX}_{\mathrm{FSBL}}$, $\mathrm{PAMAX}_{\mathrm{FSBL}}$, $\mathrm{PASUM}_{\mathrm{FSBL}}$ compare two distributions by comparing the distributions projected along some set of directions. The $\mathrm{PMAX}_{\mathrm{FSBL}}$, $\mathrm{PAMAX}_{\mathrm{FSBL}}$, and $\mathrm{PASUM}_{\mathrm{FSBL}}$ lower bounds make stronger use of a distribution than simply reducing it to its average point, so there is hope that the these bounds will help when the CBOX bound is ineffective. In queries C.2.3, C.2.4, and C.2.5, the projection-based lower bounds prune far more EMD calculations than the CBOX bound. However, pruning a large number of EMD calculations does *not* guarantee a smaller query time than achievable by brute force because of the overhead of computing a lower bound when it fails to prune an EMD calculation. In all the random partial queries C.2.*, the query times for $\mathrm{PMAX}_{\mathrm{FSBL}}$, $\mathrm{PAMAX}_{\mathrm{FSBL}}$, and $\mathrm{PASUM}_{\mathrm{FSBL}}$ were less than the query times for brute force processing, except for the $\mathrm{PMAX}_{\mathrm{FSBL}}$ and $\mathrm{PAMAX}_{\mathrm{FSBL}}$ bounds in query C.2.4. In particular, the $\mathrm{PASUM}_{\mathrm{FSBL}}$ bound performed very well for all the queries. Since the projection-based lower bounds are more expensive to compute than the CBOX lower bound, they must prune more exact EMD calculations than CBOX in order to be as effective in query time.

The queries in the final two examples of this section are both images in the Corel database. The results of the queries

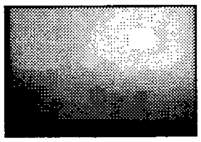
(a)

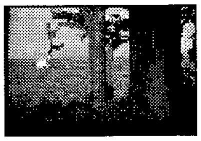


(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	15.768
CBOX	19622	0.535
PMAX$_{\text{FSBL}}$	19635	1.522
PAMAX$_{\text{FSBL}}$	19548	1.062
PASUM$_{\text{FSBL}}$	18601	1.847

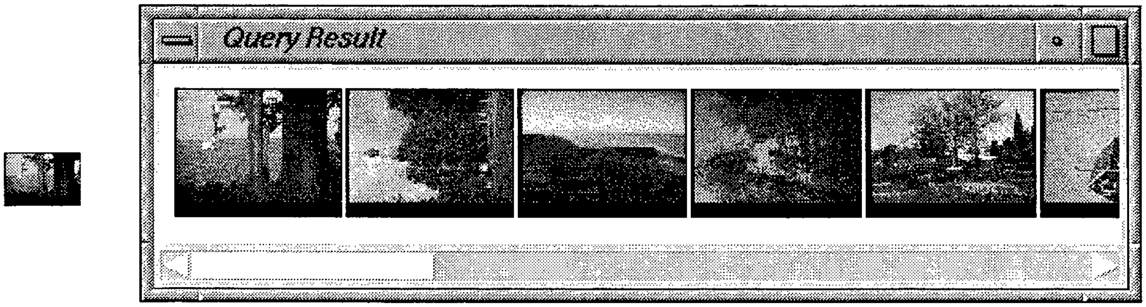
Figure 13: Query C.3.1 – sunset image. (a) query results. (b) query statistics.

C.3.1  and

C.3.2 

are shown in figure 13 and figure 14, respectively. The distributions for queries C.3.1 and C.3.2 contain 12 and 13 points, respectively. Notice that the brute force query time for the C.3.* queries is much greater than the brute force query time for the C.1.* and C.2.* queries. The difference is that both the query and the database images have a "large" number of points for the C.3.* queries. All the lower bounds perform well for query C.3.1, but the CBOX lower bound gives the lowest query time. Recall that the CBOX lower bound reduces to the distance between distribution centroids for equal-weight distributions. The centroid distance pruned many exact EMD calculations for C.3.1 because most of the weight in the distribution is around yellow and orange, far from the centroids of the database images (as one can see in figure 12). The blue, green, and brown in query C.3.2 span a larger part of the color space than the colors in C.3.1, the query centroid is close to many database centroids (once again, see figure 12), and the centroid distance lower bound does not perform as well as for C.3.1. The projection-based lower bounds, however, each give a better query time for query C.3.2 than the centroid-distance bound. Recall that the lower bounds PMAX$_{\text{FSBL}}$, PAMAX$_{\text{FSBL}}$, and PASUM$_{\text{FSBL}}$ reduce to the stronger lower bounds PMAX, PAMAX, and PASUM for equal-weight distributions. The PASUM$_{\text{FSBL}}$ lower bound yields a tolerable query time for query C.3.2.

(a)



(b)

Lower Bound	# Pruned	Query Time (s)
NONE	0	14.742
CBOX	9571	8.106
$\text{PMAX}_{\text{FSBL}}$	15094	5.893
$\text{PAMAX}_{\text{FSBL}}$	13461	6.741
$\text{PASUM}_{\text{FSBL}}$	17165	3.343

Figure 14: Query C.3.2 – image with trees, grass, water, and sky. (a) query results. (b) query statistics.

7 The EMD under Translation

Given a distribution $y = (Y, u) \in \mathbf{D}^{d,n}$, let $y \oplus t \in \mathbf{D}^{d,n}$ denote the translation of y by $t \in \mathbf{R}^d$:

$$y \oplus t = \{\ (y_1 + t, u_1), (y_2 + t, u_2), \ldots, (y_n + t, u_n)\ \}.$$

The EMD under translation $\text{EMD}_{\mathcal{T}}(x, y)$ is defined as

$$\text{EMD}_{\mathcal{T}}(x, y) = \min_{t \in \mathbf{R}^d} \text{EMD}(x, y \oplus t).$$

If

$$h^d(F, t) = \text{WORK}(F, x, y \oplus t) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(x_i, y_j + t),$$

then

$$\text{EMD}^d_{\mathcal{T}}(x, y) = \frac{\min_{t \in \mathbf{R}^d, F \in \mathcal{F}(x,y)}\ h^d(F, t)}{\min(w_\Sigma, u_\Sigma)}. \tag{15}$$

Note that $\text{EMD}^d_{\mathcal{T}}(x, y)$ is invariant under translation of x or y if $d(x_i, y_j + t) = d(x_i - t, y_j)$. Here we have added the superscript d to $\text{EMD}_{\mathcal{T}}$ to show the dependence on the ground distance function. We have also used the fact that $\mathcal{F}(x, y) = \mathcal{F}(x, y \oplus t)$, which follows directly from the fact that the weights of $y \oplus t$ are the same as the weights of y. Clearly, it suffices to minimize $h^d(F, t)$ to compute the EMD under translation. In section 7.1, we give a direct, but inefficient, algorithm to compute the global minimum of $h^d(F, t)$ over the region

$$R(x, y) = \{\ (F, t)\ :\ F \in \mathcal{F}(x, y),\ t \in \mathbf{R}^d\ \} = \mathcal{F}(x, y) \times \mathbf{R}^d.$$

In section 7.2, we give an efficient iterative algorithm that always converges monotonically, although not necessarily to the global minimum. Nonetheless, it may find the global minimum if the iteration is run with a few different initial translations.

Both the direct and iterative algorithms require a solution to the following minimization problem: for $F = (f_{ij}) \in \mathcal{F}(x, y)$ fixed, compute

$$\min_{t \in \mathbf{R}^d} h^d(F, t) = \min_{t \in \mathbf{R}^d} \text{WORK}(F, x, y \oplus t). \tag{16}$$

If

$$d(x_i, y_j + t) = d(x_i - y_j, t), \tag{17}$$

then (16) can be written as

$$\min_{t \in \mathbf{R}^d} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(x_i - y_j, t).$$

Note that condition (17) holds for any L_p distance function d. If we let $z_{ij} = x_i - y_j$ and we convert the two-dimensional index ij into a one-dimensional index l to obtain f_l and z_l, then

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(x_i, y_j + t) = \sum_{l=1}^{mn} f_l d(z_l, t),$$

and the minimization problem

$$\min_{t \in \mathbf{R}^d} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(x_i - y_j, t) = \min_{t \in \mathbf{R}^d} \sum_{l=1}^{mn} f_l d(z_l, t) \tag{18}$$

asks for a point t which minimizes a sum of weighted distances to a given set of points. This *minisum* problem is the subject of section 8, where we show how to solve the problem when the distance function d is the L_2-distance squared (section 8.1), the L_1-distance (section 8.2), and the L_2-distance (section 8.3). The solutions to these three problems allow us to compute $\text{EMD}_{\mathcal{T}}^{L_2^2}$, $\text{EMD}_{\mathcal{T}}^{L_1}$, and $\text{EMD}_{\mathcal{T}}^{L_2}$, respectively. It should be noted, however, that even for equal-weight distributions, using the L_2-distance squared for the ground distance means that the EMD is no longer a metric. One reason to consider the L_2-distance squared is that there is a simple closed form solution for the optimal translation if the distributions are equal weight (see section 8.1).

7.1 A Direct Algorithm

The function $h^d(F, t)$ is linear in F. It follows that for t fixed, the minimum value

$$\min_{F \in \mathcal{F}(x,y)} h^d(F, t)$$

is achieved at one of the vertices (dependent on t) of the convex polytope $\mathcal{F}(x, y)$. If we let

$$V(x, y) = \{ v_1, \ldots, v_N \}$$

denote the finite set of vertices of $\mathcal{F}(x, y)$, then

$$\min_{F \in \mathcal{F}(x,y)} h^d(F, t) = h^d(F^*(t), t) \qquad \text{for some vertex } F^*(t) \in V(x, y),$$

and

$$\min_{(F,t) \in R(x,y)} h^d(F, t) = \min_{t \in \mathbf{R}^d} h^d(F^*(t), t). \tag{19}$$

The minimum on the right-hand side of (19) can be rewritten as

$$\min_{t\in\mathbf{R}^d} h^d(F^*(t),t) = \min_{F\in V(x,y)} \min_{t\in\mathbf{R}^d} h^d(F,t),$$

so that

$$\min_{(F,t)\in R(x,y)} h^d(F,t) = \min_{F\in V(x,y)} \min_{t\in\mathbf{R}^d} h^d(F,t). \tag{20}$$

Thus, if the innermost minimum on the right-hand side of (20) exists, then the minimum on the left-hand side of (20) must also exist and must be achieved at some (F^*, t^*), where $F^* \in V(x,y)$. Given an algorithm to compute

$$\min_{t\in\mathbf{R}^d} h^d(F,t)$$

for a fixed F, the minimum on the left-hand side of (20) may be computed by simply looping over all the vertices in $V(x,y)$:

$$\min_{(F,t)\in R(x,y)} h^d(F,t) = \min_{k=1,\ldots,N} \min_{t\in\mathbf{R}^d} h^d(v_k,t). \tag{21}$$

Only a finite number of flow values must be examined to find the minimum work.

Although this simple strategy guarantees that we find a globally optimal translation, it is not practical because N can be very large. We may eliminate the variable f_{11} in the definition of a feasible flow by solving (4) for f_{11} as an affine combination of the other f_{ij}'s. Substituting for f_{11} in (1), (2), and (3) leaves $mn + m + n$ linear inequalities. This reasoning shows that $\mathcal{F}(x,y)$ is an $(mn-1)$-dimensional convex polytope defined by the intersection of $mn + m + n$ halfspaces. The Upper Bound Theorem ([13],[3]) states that a simple polytope in $\mathbf{R}^d$ with n facets has $O(n^{\lfloor d/2 \rfloor})$ vertices, and there are examples for which this bound is tight. Therefore, $\mathcal{F}(x,y)$ can have as many as $N = O((mn-1)^{mn+m+n})$ vertices. Even for small values of m and n, this is too many vertices to exhaustively check in a reasonable amount of time. The beauty of the simplex algorithm ([10]) for solving a linear program is that it provides a method for visiting vertices of the feasible polytope in such a way that the objective function always gets closer to its optimal value (and the number of vertices visited is always no larger in order than the maximum of the number of variables and the number of constraints). In the next section, we give an iterative algorithm that generates a sequence of (flow,translation) pairs for which the amount of work decreases or remains constant at every step.

7.2 An Iterative Algorithm

Consider the following iteration that begins with an initial translation $t^{(0)}$:

$$F^{(k)} = \left(f_{ij}^{(k)}\right) = \arg\left(\min_{F=(f_{ij})\in\mathcal{F}(x,y)} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d\left(x_i, y_j + t^{(k)}\right)\right), \tag{22}$$

$$t^{(k+1)} = \arg\left(\min_{t\in\mathbf{R}^d} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d(x_i, y_j + t)\right). \tag{23}$$

The minimization problem on the right-hand side of (22) is the familar transportation problem. Under the assumption (17), the minimization problem on the right-hand side of (23) is the minisum problem (18) to be covered in section 8. The flow and translation iterates define the work and EMD iterates

$$\begin{aligned} \mathrm{WORK}^{(k)} &= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, y_j + t^{(k)}\right) = \mathrm{WORK}\left(F^{(k)}, x, y \oplus t^{(k)}\right), \\ \mathrm{EMD}^{(k)} &= \frac{\mathrm{WORK}^{(k)}}{\min(w_\Sigma, u_\Sigma)}. \end{aligned}$$

The order of evaluation is

$$\underbrace{t^{(0)} \longrightarrow F^{(0)}}_{\mathrm{WORK}^{(0)},\, \mathrm{EMD}^{(0)}} \longrightarrow \underbrace{t^{(1)} \longrightarrow F^{(1)}}_{\mathrm{WORK}^{(1)},\, \mathrm{EMD}^{(1)}} \longrightarrow \cdots .$$

By (22), we have

$$\mathrm{WORK}^{(k+1)} = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k+1)} d\left(x_i, y_j + t^{(k+1)}\right) \leq \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, y_j + t^{(k+1)}\right). \tag{24}$$

From (23), we know

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, y_j + t^{(k+1)}\right) \leq \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, y_j + t^{(k)}\right) = \mathrm{WORK}^{(k)}. \tag{25}$$

Combining (24) and (25) shows

$$\mathrm{WORK}^{(k+1)} \leq \mathrm{WORK}^{(k)}. \tag{26}$$

The decreasing sequence $\left(\mathrm{WORK}^{(k)}\right)$ is bounded below by zero, and hence it converges ([7]). There is, however, no guarantee that the work iteration converges to the global minimum of $h^d(F,t) = \mathrm{WORK}(F, x, y \oplus t)$.

One way for the work iteration to converge is if $F^{(k)}$ is returned in step (22) as an optimal flow for $t^{(k)}$, and $t^{(k+1)} = t^{(k)}$ is returned in step (23) as an optimal translation for $F^{(k)}$. Denote the indicator function for this event as $\mathrm{MUTUAL}\left(F^{(k)}, t^{(k)}\right)$. It is clear that

$$\mathrm{MUTUAL}\left(F^{(k)}, t^{(k)}\right) \Rightarrow \left\{ \begin{array}{ccccc} t^{(k)} & = & t^{(k+1)} & = & \cdots , \\ F^{(k)} & = & F^{(k+1)} & = & \cdots , \\ \mathrm{WORK}^{(k)} & = & \mathrm{WORK}^{(k+1)} & = & \cdots . \end{array} \right. \quad \text{and}$$

The fact that $F^{(k)}$ is an optimal flow for $t^{(k)}$ implies

$$\frac{\partial h^d}{\partial F}\left(F^{(k)}, t^{(k)}\right) = 0, \tag{27}$$

where a neighborhood of $F \in \partial(\mathcal{F}(x,y))$ must be restricted to lie within $\mathcal{F}(x,y)$. The fact that $t^{(k)}$ is an optimal translation for $F^{(k)}$ implies

$$\frac{\partial h^d}{\partial t}\left(F^{(k)}, t^{(k)}\right) = 0. \tag{28}$$

Combining conditions (27) and (28) shows that the work iteration converges to either a local minimum or a saddle point value if MUTUAL $\left(F^{(k)}, t^{(k)}\right)$ is true.

Now suppose that the routine that solves the linear program (LP) in (22) always returns a vertex of $\mathcal{F}(x, y)$. The simplex algorithm, for example, always returns a vertex of the feasible polytope. This is possible since there is always a vertex of the feasible polytope at which a linear objective function achieves its minimum. With the assumption that the flow iterates are always vertices of $\mathcal{F}(x, y)$, there will be only a finite number of points (F, t) that the work iteration visits because there are a finite number of flow iterates, and each translation iterate (other than the initial translation) must be an optimal translation returned for one of the flow iterates. It follows that there are only a finite number of work values generated. Since the work iteration is guaranteed to converge, the work iterates must stabilize at one of these work values. Suppose

$$\mathrm{WORK}^{(k)} = \mathrm{WORK}^{(k+1)} = \cdots . \tag{29}$$

Since there are only a finite number of pairs (F, t) visited, condition (29) implies that there must be a repeating cycle of pairs:

$$\left(F^{(k)}, t^{(k)}\right), \ \ldots \ , \left(F^{(k+r-1)}, t^{(k+r-1)}\right), \left(F^{(k+r)}, t^{(k+r)}\right) = \left(F^{(k)}, t^{(k)}\right), \ \ldots .$$

For $r > 1$, the work iteration converges even though the flow and translation iterations do not converge. However, such a non-trivial (flow,translation) cycle is unstable in the sense that it can be broken (for any real problem data) by perturbing one of the translation iterates by a small amount. In practice, the work iteration almost always converges because a length $r = 1$ cycle occurs. A cycle of length $r = 1$ starting at $\left(F^{(k)}, t^{(k)}\right)$ is exactly the condition MUTUAL $\left(F^{(k)}, t^{(k)}\right)$, and we previously argued that the work iteration converges to a critical value in this case.

Finally, let us show that the work sequence will stabilize at the global minimum once $F^{(k)} = F^*$, where (F^*, t^*) is optimal for some t^*. First, it is easy to see that if $(F^*, t^*) = \left(F^{(k)}, t^{(k)}\right)$ is optimal, then $h^d(F^*, t^*) = \mathrm{WORK}^{(k)} = \mathrm{WORK}^{(k+1)} = \cdots$. This is an immediate consequence of the optimality of (F^*, t^*) and the monotonicity condition (26). Now suppose $F^{(k)} = F^*$, where (F^*, t^*) is optimal. Note that $t^{(k+1)}$ and t^* both solve (23), so

$$h^d\left(F^*, t^{(k+1)}\right) = h^d\left(F^{(k)}, t^{(k+1)}\right) = h^d\left(F^{(k)}, t^*\right) = h^d(F^*, t^*).$$

(If (23) has a unique solution, then $t^{(k+1)} = t^*$.) Since condition (24) gives

$$h^d\left(F^{(k+1)}, t^{(k+1)}\right) \leq h^d\left(F^{(k)}, t^{(k+1)}\right) = h^d(F^*, t^*),$$

and since

$$h^d\left(F^{(k+1)}, t^{(k+1)}\right) \geq h^d(F^*, t^*) \qquad \text{(optimality of } (F^*, t^*)\text{)},$$

we must have

$$\mathrm{WORK}^{(k+1)} = h^d\left(F^{(k+1)}, t^{(k+1)}\right) = h^d(F^*, t^*).$$

(If (22) has a unique solution, then $F^{(k+1)} = F^*$.) We have already argued that once the work sequence hits the minimum, it must repeat at this minimum forever.

Let us summarize the results of this section. The work iteration always converges. We can arrange to have all flow iterates at the vertices of $\mathcal{F}(x, y)$. In this case, the (flow,translation) iterates must cycle. A cycle of length $r > 1$ will almost never occur, and a cycle of length $r = 1$

implies that the (flow,translation) sequence converges to a critical point and, therefore, that the work sequence converges to either a local minimum or a saddle point value. Thus, in practice the work iteration almost always converges to a critical value. If the flow iteration ever reaches a vertex at which the minimum work occurs with a suitable choice of translation, then the work iteration converges to the global minimum. Since there is no guarantee that the work iteration converges to the global minimum, the iterations should be run with a few different starting translations $t^{(0)}$ in search of the true minimum work. In some preliminary experiments, we have found that the work iteration usually converges within a handful of iterations (three to five) using d equal to the L_2-distance squared, the L_1-distance, or the L_2-distance.

8 Minimizing a Weighted Sum of Distances

The abstract minimization problem considered in this section is

$$\min_p \sum_{i=1}^{n} w_i d(p, p_i).$$

We now show how to solve this problem when d is the L_2-distance squared, the L_1-distance, and the L_2-distance.

8.1 Minimizing a Weighted Sum of Squared L_2 Distances

If d is the L_2-distance squared, then the minisum problem is a weighted sum of squares problem

$$\min_p \sum_{i=1}^{n} w_i ||p - p_i||_2^2.$$

It is well-known (and easily proven using standard calculus) that the unique optimal location p^* is at the centroid

$$p^* = \overline{p} = \frac{\sum_{i=1}^{n} w_i p_i}{w_\Sigma}.$$

Returning the original problem (18) for a moment, we have

$$\begin{aligned} t^* = \overline{z} &= \frac{\sum_{l=1}^{mn} f_l z_l}{\sum_{l=1}^{mn} f_l} \\ &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}(x_i - y_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \\ \overline{z} &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}(x_i - y_j)}{\min(w_\Sigma, u_\Sigma)}. \end{aligned}$$

When x and y are equal-weight distributions,

$$\begin{aligned} \overline{z} &= \frac{\sum_{i=1}^{m} x_i \sum_{j=1}^{n} f_{ij}}{w_\Sigma} - \frac{\sum_{j=1}^{n} y_j \sum_{i=1}^{m} f_{ij}}{u_\Sigma} \\ &= \frac{\sum_{i=1}^{m} w_i x_i}{w_\Sigma} - \frac{\sum_{j=1}^{n} u_j y_j}{u_\Sigma} \qquad ((6),(7)) \\ t^* = \overline{z} &= \overline{x} - \overline{y}. \end{aligned}$$

In the equal weight case, the best translation for any feasible flow $F = (f_{ij})$ is $\overline{x} - \overline{y}$. The iteration given in section 7.2 is not needed in this case to compute $\mathrm{EMD}_{\mathcal{T}}^{L_2^2}(x, y)$. Instead, simply translate y by $\overline{x} - \overline{y}$ (this lines up the centroids of x and y) and compute $\mathrm{EMD}^{L_2^2}(x, y \oplus (\overline{x} - \overline{y}))$.

8.2 Minimizing a Weighted Sum of L_1 Distances

In this section, we consider the minisum problem when d is the L_1-distance. The minimization problem is

$$\begin{aligned}
\min_p \sum_{i=1}^n w_i\|p-p_i\|_1 &= \min_p \sum_{i=1}^n w_i \sum_{k=1}^d \left|p^{(k)} - p_i^{(k)}\right| \\
&= \min_p \sum_{k=1}^d \left(\sum_{i=1}^n w_i \left|p^{(k)} - p_i^{(k)}\right| \right) \\
\min_p \sum_{i=1}^n w_i\|p-p_i\|_1 &= \sum_{k=1}^d \left(\min_{p^{(k)}} \sum_{i=1}^n w_i \left|p^{(k)} - p_i^{(k)}\right| \right),
\end{aligned}$$

where $p^{(k)}$ and $p_i^{(k)}$ are the kth components of p and p_i, respectively. Thus, a solution to the problem in one dimension gives a solution to the problem in d dimensions by simply collecting the optimal location for each of the one-dimensional problems into a d-dimensional vector.

Now suppose $p_1 < p_2 < \cdots < p_n$ are points along the real line, and we want to minimize

$$g(p) = \sum_{i=1}^n w_i|p-p_i|.$$

Let $p_0 = -\infty$ and $p_{n+1} = +\infty$. Then

$$g(p) = \sum_{i=1}^l w_i(p-p_i) + \sum_{i=l+1}^n w_i(p_i - p) \qquad \text{for } p \in [p_l, p_{l+1}],\ \ l = 0, \ldots, n.$$

Over the interval $[p_l, p_{l+1}]$, $g(p)$ is affine in p:

$$g(p) = \left(\sum_{i=1}^l w_i - \sum_{i=l+1}^n w_i \right) p + \left(\sum_{i=l+1}^n w_i p_i - \sum_{i=1}^l w_i p_i \right) \qquad \text{for } p \in [p_l, p_{l+1}].$$

If we let

$$m_l = \sum_{i=1}^l w_i - \sum_{i=l+1}^n w_i \tag{30}$$

denote the slope of $g(p)$ over $[p_l, p_{l+1}]$, then

$$-w_\Sigma = m_0 < m_1 < \cdots < m_n = w_\Sigma,$$

and

$$m_{l+1} = m_l + 2w_l.$$

The function $g(p)$ is a continuous piecewise linear function with slope increasing from a negative value at $-\infty$ to a positive value at $+\infty$, and as such it obviously has a minimum value at the point when its slope first becomes nonnegative. Let

$$l^* = \min \{\, l \ :\ m_l \geq 0 \,\}.$$

If $m_{l^*} \neq 0$, then then the unique minimum value of $g(p)$ occurs at p_{l^*}. Otherwise, $m_{l^*} = 0$ and the minimum value of $g(p)$ is achieved for $p \in [p_{l^*}, p_{l^*+1}]$. See figure 15. In the special case of

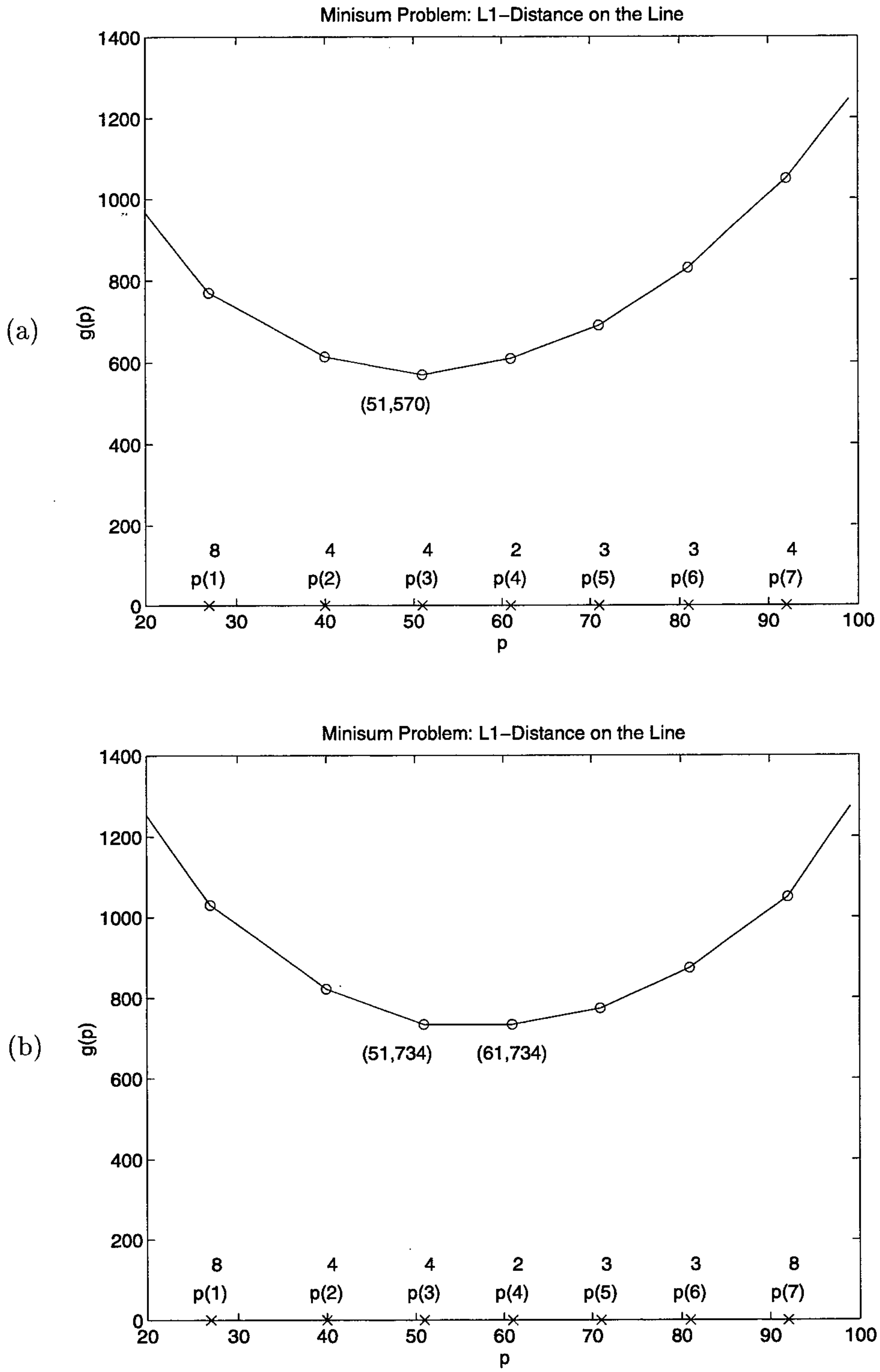


Figure 15: The minisum problem on the line with unequal weights. (a) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [8, 4, 4, 2, 3, 3, 4]$: $l^* = 3$, $m_{l^*} > 0$, and there is a unique minimum at $p_3 = 51$. (b) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [8, 4, 4, 2, 3, 3, 8]$: $l^* = 3$, $m_{l^*} = 0$, and the minimum occurs at every value in $[p_3, p_4] = [51, 61]$.

equal-weight points, the minimum value occurs at the ordinary median value of the points. If $w_i \equiv w$, then it follows easily from (30) that $m_l = w(2l - n)$. If n is odd, then $l^* = \lceil n/2 \rceil$, $m_{l^*} > 0$, and the unique minimum of $g(p)$ occurs at the median point $p_{\lceil n/2 \rceil}$. If n is even, then $l^* = n/2$, $m_{l^*} = 0$, and the minimum value of $g(p)$ is attained for every point in the interval $[p_{n/2}, p_{(n/2)+1}]$. See figure 16.

8.3 Minimizing a Weighted Sum of L_2 Distances

The final minisum problem that we consider is when d is the L_2-distance function. The minimization problem

$$\min_p \sum_{i=1}^{n} w_i ||p - p_i||_2 \tag{31}$$

has a long history ([15]). A basic iteration procedure that solves this problem was proposed in 1937 by Weiszfeld ([14]). Consider the objective function

$$g(p) = \sum_{i=1}^{n} w_i ||p - p_i||_2.$$

If the points $p_1, \ldots, p_n$ are not collinear, then $g(p)$ is strictly convex and has a unique minimum. If $p_1, \ldots, p_n$ are collinear, then an optimal point must lie on the line through the given points (if not, one could project the claimed optimal point onto to the line, thereby decreasing its distance to all the given points, to obtain a better point). In this case, the algorithm given in section 8.2 for points on the real line can be used (the L_2-distance reduces to the absolute value in one-dimension). The objective function is differentiable everywhere except at the given points:

$$\frac{\partial g}{\partial p} = \sum_{i=1}^{n} \frac{w_i(p - p_i)}{||p - p_i||_2}.$$

Setting the partial derivative to zero results in the equation

$$\sum_{i=1}^{n} \frac{w_i(p - p_i)}{||p - p_i||_2} = 0,$$

which cannot be solved explicitly for p. The Weiszfeld iteration replaces the p in the numerator by the $(k+1)$st iterate $p^{(k+1)}$ and the p in the denominator by the kth iterate $p^{(k)}$, and solves for $p^{(k+1)}$:

$$p^{(k+1)} = \begin{cases} \frac{\sum_{i=1}^{n} w_i ||p^{(k)} - p_i||_2^{-1} p_i}{\sum_{i=1}^{n} w_i ||p^{(k)} - p_i||_2^{-1}} & \text{if } p^{(k)} \neq p_1, \ldots, p_n \\ p_i & \text{if } p^{(k)} = p_i \end{cases} .$$

Here are some facts about this iteration (assuming the input points are not collinear).

- The iteration always converges. ([9])
- If no iterate $p^{(k)}$ is equal one of the given points, then the iteration converges to the global minimum location of $g(p)$. ([9])
- The iteration can fail to converge to the global minimum location for a continuum of starting values $p^{(0)}$ because some iterate $p^{(k)}$ becomes equal to a non-optimal given point. ([2])
- If the optimal location is *not* at one of the given points, then convergence will be linear. ([8])

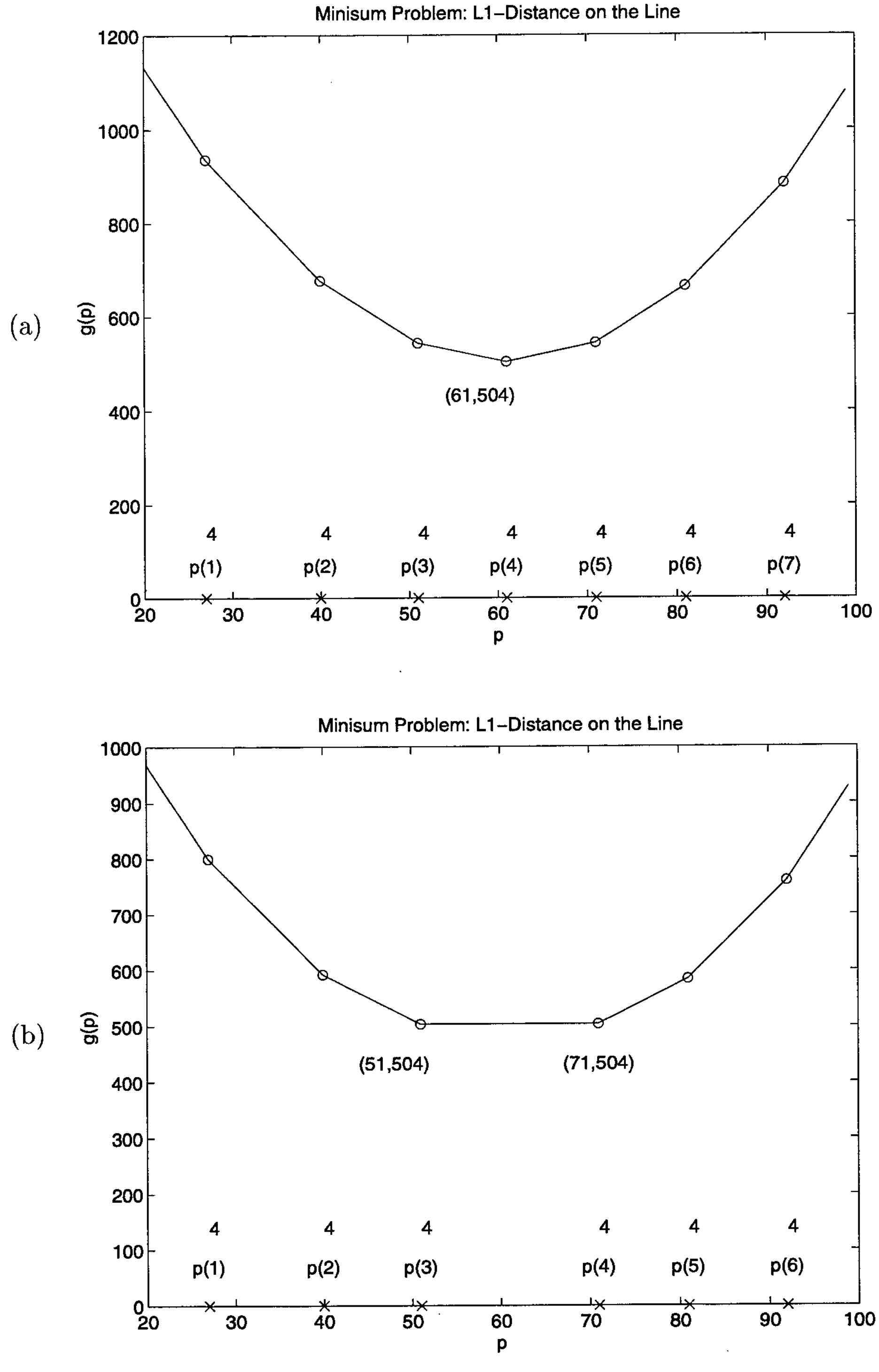


Figure 16: The minisum problem on the line with equal weights. (a) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [4, 4, 4, 4, 4, 4, 4]$: $l^* = 4$, $m_{l^*} > 0$, and there is a unique minimum at the ordinary median $p_4 = 61$. (b) $p = [27, 40, 51, 71, 81, 92]$, $w = [4, 4, 4, 4, 4, 4]$: $l^* = 3$, $m_{l^*} = 0$, and the minimum occurs at every value in the interval $[p_3, p_4] = [51, 71]$.

- If the optimal location *is* at one of the given points, then convergence can be linear, superlinear, or sublinear. ([8])

Since convergence to the global minimum location is not guaranteed, the iteration should be run more than once with different starting points.

It is conjectured in [2] that if the starting point is within the affine subspace P spanned by the given points, then the Weiszfeld iteration is guaranteed to converge to the global minimum location for all but a finite number of such starting points. If this conjecture is true, then the iteration will converge with high probability to the optimal location if one chooses a random starting point in P. Note that P is the entire space $\mathbf{R}^d$ if the $n-1$ vectors $p_n - p_1, p_n - p_2, \ldots, p_n - p_{n-1}$ span all of $\mathbf{R}^d$. If the given points are random, this event is very likely to occur if $n-1 \geq d$. In regards to speeding up convergence, see [5] for an accelerated Weiszfeld procedure.

9 Conclusion

We have presented several lower bounds on the EMD which do not require equal-weight distributions, and are therefore applicable to partial queries. The effectiveness of the bounds was illustrated in a color-based retrieval system where applying one bound per query almost always resulted in a smaller query time than brute force query processing. Using a combination of bounds per query may improve search times even more. In particular, a promising combination seems to be the CBOX bound followed by the $\text{PASUM}_{\text{FSBL}}$ projection-based bound. The CBOX bound is faster to compute, but the $\text{PASUM}_{\text{FSBL}}$ bound makes stronger use of the distributions than simply using averages. The latter bound seems to be the best of the projection-based bounds that we proposed, although this may vary with the database and mode of query. More experimentation is needed to see if there is a clear best bound or combination of bounds for a majority of applications.

The other main topic of this work was computing the EMD under translation. The frameworks of the proposed methods are still applicable when the transformation group is not the translation group. In our methods, we must solve the problem of finding the best transformation for a given flow. This problem reduces to problems with known solutions in the translation case when the ground distance is the L_1-distance, the L_2-distance, or the L_2-distance squared. Once we can find the best transformation for a given flow, we can still find the global minimum by looping over the vertices of a convex polytope, and a local minimum (almost always) using our two stage minimization framework. Future work will consider other types of transformations such as Euclidean, similarity, and affine transformations.

Acknowledgements

We would like to thank Yossi Rubner for his transportation problem code and for the color signatures of the Corel database images used in our experiments.

References

[1] M. Bern, D. Eppstein, L. Guibas, J. Hershberger, S. Suri, and J. Wolter. The centroid of points with approximate weights. In *Proceedings of Third Annual European Symposium on Algorithms*, pages 460–472, 1995.

[2] R. Chandrasekaran and A. Tamir. Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. *Mathematical Programming, Series A*, 44(3):293–295, Nov. 1989.

[3] K. L. Clarkson. A bound on local minima of arrangements that implies the upper bound theorem. *Discrete & Computational Geometry*, 10(4):427–433, 1993.

[4] G. B. Dantzig. Application of the simplex method to a transportation problem. In *Activity Analysis of Production and Allocation*, pages 359–373. John Wiley and Sons, 1951.

[5] Z. Drezner. A note on the Weber location problem. *Annals of Operations Research*, 40(1–4):153–161, 1992.

[6] F. S. Hillier and G. J. Lieberman. *Introduction to Mathematical Programming*, pages 202–229. McGraw-Hill, 1990.

[7] R. Johnsonbaugh and W. E. Pfaffenberger. *Foundations of Mathematical Analysis*, pages 49–50. Marcel Dekker, inc., 1981.

[8] I. N. Katz. Local convergence in Fermat's problem. *Mathematical Programming*, 6(1):89–104, Feb. 1974.

[9] H. W. Kuhn. A note on Fermat's problem. *Mathematical Programming*, 4:98–107, 1973.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*, pages 430–443. Cambridge University Press, second edition, 1992.

[11] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the APRA Image Understanding Workshop*, pages 661–668, May 1997.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision*, 1998. To appear.

[13] R. Seidel. The upper bound theorem for polytopes: An easy proof of its asymptotic version. *Computational Geometry: Theory and Applications*, 5(2):115–116, Sept. 1995.

[14] E. V. Weiszfeld. Sur le point par lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematics Journal*, 43:355–386, 1937.

[15] G. O. Wesolowsky. The weber problem: History and perspectives. *Location Science*, 1(1):5–23, May 1993.

[16] G. Wyszecki and W. S. Styles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 1982.

I A Lower Bound on the L_2-Norm in terms of the L_1-Norm

Theorem 7

$$||a||_2 \geq \frac{1}{\sqrt{d}}||a||_1 \qquad \forall\, a \in \mathbf{R}^d.$$

Proof The inequality obviously holds when $a = 0$, so it suffices to show that

$$\min_{a \neq 0} \frac{||a||_2}{||a||_1} = \frac{1}{\sqrt{d}}.$$

The homogeneity of all L_p norms

$$||ca||_p = |c|\,||a||_p \quad \text{for } c \in \mathbf{R}$$

implies that

$$\min_{a \neq 0} \frac{||a||_2}{||a||_1} = \min_{||a||_1 = 1} ||a||_2.$$

If abs(a) denotes the vector formed by taking the absolute value of each of the components of a, then $||\text{abs}(a)||_p = ||a||_p$. It follows that

$$\min_{||a||_1=1} ||a||_2 = \min_{a \geq 0,\, ||a||_1 = 1} ||a||_2.$$

Let

$$f(a) = \sum_{k=1}^{d} a_k^2 \qquad \text{and} \qquad g(a) = \left(\sum_{k=1}^{d} a_k\right) - 1.$$

Then

$$\min_{a \geq 0,\, ||a||_1=1} ||a||_2 = \left(\min_{g(a)=0} f(a)\right)^{\frac{1}{2}}.$$

According to the theory of Lagrange multipliers, we must have

$$\begin{aligned} (\nabla f)(a^*) &= \lambda((\nabla g)(a^*)) \quad \text{for some } \lambda \in \mathbf{R}. \\ 2a^* &= \lambda 1 \end{aligned}$$

at an extremum location a^*, where 1 denotes a vector of d ones. Solving for a^* gives $a_k^* = \lambda/2$ for $k = 1, \ldots, d$. Solving for λ in the constraint $g(a^*) = 0$ gives $\lambda = 2/d$. Hence $a_k^* = 1/d$ for $k = 1, \ldots, d$, and $f(a^*) = 1/d$. Obviously, there is no maximum value for the homogeneous function $||a||_2$ on the plane $g(a) = 0$. Therefore,

$$\min_{g(a)=0} f(a) = f(a^*) = \frac{1}{d}.$$

Taking the square root of both sides completes the proof. ■