Chapter 6

The EMD under Transformation Sets

A major challenge in image retrieval applications is that the images we desire to match can be visually quite different. This can happen even if these images are views of the same scene because of illumination changes, viewpoint motion, occlusions, etc.. Consider for example, recognizing objects by their color signatures. A direct comparison of color histograms or an EMD between color signatures of imaged objects does not account for lighting differences. In [28], Healey and Slater show that an illumination change results in a linear transformation of the image pixel colors (under certain reasonable assumptions). In a similar result, sensor measurements of multispectral satellite images recorded under different illumination and atmospheric conditions differ by an affine transformation ([27]).

The general problem of comparing features modulo some transformation set also arises in texture-based and shape-based image retrieval. In [69], the texture content signature of a single texture image is a collection of spatial frequencies, where each frequency is weighted by the amount of energy at that frequency. If the frequency features are represented in log-polar coordinates, then scaling the texture results in a feature translation along the log-scale axis, while rotating the texture results in a feature translation along the cyclic orientation axis. These translations must be taken into account by a texture distance measure which is invariant to scaling and/or rotation.

For shape-based retrieval, suppose we summarize the shape content of an image as a collection of curves or feature points in the image. Then changes in viewpoint and/or viewing distances will result in changes in the coordinates of the extracted features, even though we are looking at an image of the same scene. Allowing for a transformation is necessary, for example, in matching point features in stereo image pairs. Even if the viewpoint and viewing distance for two images of the same scene are roughly the same, the images may have been acquired at different resolutions or drawn with different drawing programs which use different units and have different origin points. Direct comparison of summary feature coordinates is not likely to capture the visual similarity between the underlying images.

The *Earth Mover's Distance under a transformation set* is the minimum EMD between one distribution and a transformed version of the other distribution, where transformations are chosen from a given set. The allowable transformations of a distribution are dictated by the application. Sets of distribution transformations can be divided into three classes: those with transformations that change (I) only the weights, (II) only the points, and (III) both the weights and the points of a distribution. This chapter focuses on class (II) transformation sets. The previously mentioned applications of lighting-invariant object recognition, scale and orientation-invariant texture retrieval, and point feature matching in stereo image pairs use class (II) sets.

Some applications may call for class (III) sets. Suppose, for example, that a distribution point captures the location and properties of an image region, and that its corresponding weight is the region area. The EMD between two such distributions implicitly defines similar images as those in which regions of similar size and properties are close to one another. This measure will not capture visual similarities present at different scales within two images unless we allow for a transformation of both region locations and areas. Such transformations change both the points and weights of a distribution.

We have already seen an application involving a class (I) set. The scale estimation problem described in section 4.5 is formulated as the EMD under transformation (EMD$_\mathcal{G}$) problem

$$c^0 = \max \arg \boxed{\min_{g_c \in \mathcal{G}} \ \mathrm{EMD}(\mathbf{x}, g_c(\mathbf{y}))}\,, \tag{6.1}$$

where $\mathcal{G} = \{ \, g_c \, : \, g_c(\mathbf{y}) = g_c(Y,u) = (Y, cu), \; 0 < c \leq 1 \, \}$. In words, $\mathcal{G}$ consists of transformations g_c that scale down the weights of a distribution by a factor c. The EMD$_\mathcal{G}$ problem is the boxed minimization in (6.1). Analysis of the function $E(c) = \mathrm{EMD}(\mathbf{x}, (Y, cu))$ in section 4.5 revealed a lot of structure: $E(c)$ decreases as c decreases until it becomes constant for all c less than or equal to some c^0. The scale estimate is c^0, which is the largest weight scale c that minimizes the EMD between $\mathbf{x}$ and (Y, cu). Please refer back to section 4.5 for more details and an efficient solution to (6.1) which takes full advantage of the structure of $E(c)$.

This chapter is organized as follows. In section 6.1, we give some basic definitions and notation, including a formal definition of the Earth Mover's Distance under a transformation set and related measures. In section 6.2, we give a direct, but inefficient, algorithm to compute a globally optimal (flow,transformation) pair that yields the EMD under class (II) transformation sets. In section 6.3, we give an iteration for class (II) sets that always converges monotonically, although it may converge to a transformation which is only locally (as opposed to globally) optimal. The FT iteration is applicable to any transformation set for which there is an algorithm to minimize a weighted sum of distances from one point set to a transformed version of the other. This *optimal transformation problem*, which is also a required subroutine for the direct algorithm, is the subject of section 6.4. In section 6.5, we show how the FT iteration may still be applied for a useful class (III) transformation set.

In section 6.6, we consider some specific combinations of transformation set, ground distance function, and feature space for which a globally optimal transformation can be computed directly, without the aid of our iteration. We return to the FT iteration in section 6.7, where we consider questions of convergence to only a locally optimal transformation. In section 6.8, we cover two miscellaneous topics: the tradeoffs in choosing between the L_2^2 and L_2 ground distances, and the growth rate of the EMD with respect to transformation parameters. Although the former topic is discussed in the context of the $\mathrm{EMD}_{\mathcal{G}}$ problem, other criteria are also considered. Finally, in section 6.9 we apply the FT iteration to the problems of (i) illumination-invariant object recognition, and (ii) point feature matching in stereo image pairs.

In [12], we proposed the previously mentioned iteration for the case of translation.

6.1 Definitions and Notation

The *Earth Mover's Distance under transformation set* $\mathcal{G}$ is defined as

$$\mathrm{EMD}_{\mathcal{G}}(\mathbf{x},\mathbf{y}) = \min(\min_{g\in\mathcal{G}} \mathrm{EMD}(\mathbf{x}, g(\mathbf{y})), \min_{g\in\mathcal{G}} \mathrm{EMD}(g(\mathbf{x}), y)), \tag{6.2}$$

where $g(\mathbf{x})$ is the result of applying the transformation $g \in \mathcal{G}$ to the distribution $\mathbf{x}$. In the case that

(i) the transformations in $\mathcal{G}$ only modify the distribution points,

(ii) $\mathcal{G}$ is a transformation group (and therefore every element of $\mathcal{G}$ has an inverse element), and

(iii) the ground distance d satisfies $d(x, g(y)) = d(g^{-1}(x), y)$ for all $g \in \mathcal{G}$,

we have $\mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) = \mathrm{EMD}(g^{-1}(\mathbf{x}), \mathbf{y}))$, and definition (6.2) reduces to

$$\mathrm{EMD}_{\mathcal{G}}(\mathbf{x},\mathbf{y}) = \min_{g\in\mathcal{G}} \mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) = \min_{g\in\mathcal{G}} \mathrm{EMD}(g(\mathbf{x}), \mathbf{y})). \tag{6.3}$$

Condition (ii) is satisfied, for example, when $\mathcal{G} = \mathcal{T}$, the group of translations, and when $\mathcal{G} = \mathcal{E}$, the group of Euclidean transformations (rotation plus translation). It is not satisfied by the set of similarity transformations $\mathcal{S}$ (uniform scaling plus rotation plus translation), linear transformations $\mathcal{L}$, and affine transformations $\mathcal{A}$ (linear plus translation). Transformations that shrink a point set to a single point (scale parameter is zero) do not have inverses. Condition (iii) is satisfied, for example, with $\mathcal{G} = \mathcal{T}$ and ground distance d equal to any L_p distance function, and with $\mathcal{G} = \mathcal{E}$ and the ground distance d equal to the L_2 distance function.

We can combine the partial matching allowed by EMD^γ and the transformations allowed by $\mathrm{EMD}_\mathcal{G}$. The *partial Earth Mover's Distance under transformation set* $\mathcal{G}$ is defined as

$$\mathrm{EMD}_\mathcal{G}^\gamma(\mathbf{x},\mathbf{y}) = \min(\min_{g\in\mathcal{G}} \mathrm{EMD}^\gamma(\mathbf{x}, g(\mathbf{y})), \min_{g\in\mathcal{G}} \mathrm{EMD}^\gamma(g(\mathbf{x}), y)). \tag{6.4}$$

In the case that conditions (i), (ii), and (iii) hold, definition (6.4) reduces to

$$\mathrm{EMD}_\mathcal{G}^\gamma(\mathbf{x},\mathbf{y}) = \min_{g\in\mathcal{G}} \mathrm{EMD}^\gamma(\mathbf{x}, g(\mathbf{y})) = \min_{g\in\mathcal{G}} \mathrm{EMD}^\gamma(g(\mathbf{x}), \mathbf{y})). \tag{6.5}$$

Using the partial EMD under a transformation set may be useful, for example, in matching point features in stereo image pairs. The fraction parameter γ compensates for the fact that only some features appear in both images, and the set parameter $\mathcal{G}$ accounts for the appropriate transformation between corresponding features.

We shall now prove that the EMD under a transformation set is a metric when conditions (i), (ii), and (iii) are satisfied and the EMD itself is a metric. Recall that the EMD is a metric on distributions when the ground distance is a metric and the distributions have equal total weight. For a precise statement of the theorem, we need to define equivalence classes on distributions under the group $\mathcal{G}$. Two distributions $\mathbf{x}$ and $\mathbf{y}$ are in the same $\mathcal{G}$-equivalence class iff $\mathbf{x} = g(\mathbf{y})$ for some $g \in \mathcal{G}$. The equivalence class $E(\mathbf{x})$ that contains distribution $\mathbf{x}$ is

$$E(\mathbf{x}) = \{\ g(\mathbf{x}) : g \in \mathcal{G}\ \}.$$

We say that two distributions $\mathbf{x}$ and $\widehat{\mathbf{x}}$ are *$\mathcal{G}$-equivalent* iff $\mathbf{x}, \widehat{\mathbf{x}} \in E(\mathbf{x})$, and we denote this equivalence as $\mathbf{x} \sim \widehat{\mathbf{x}}$. Note that

$$\mathrm{EMD}_\mathcal{G}(\mathbf{x},\mathbf{y}) = \mathrm{EMD}_\mathcal{G}(\widehat{\mathbf{x}},\widehat{\mathbf{y}}) \quad \forall \mathbf{x} \sim \widehat{\mathbf{x}}, \mathbf{y} \sim \widehat{\mathbf{y}}.$$

The EMD under a transformation group is then well defined on $\mathcal{G}$-equivalence classes by

$$\mathrm{EMD}_{\mathcal{G}}(E_1, E_2) = \mathrm{EMD}_{\mathcal{G}}(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x} \in E_1, \mathbf{y} \in E_2. \tag{6.6}$$

Theorem 12 *The EMD under a transformation group $\mathcal{G}$ is a metric on distribution $\mathcal{G}$-equivalence classes when conditions (i), (ii), and (iii) are satisfied and the EMD itself is a metric on distributions.*

Proof. Obviously, $\mathrm{EMD}_{\mathcal{G}}(E_1, E_2) \geq 0$ for every pair of $\mathcal{G}$-equivalence classes E_1 and E_2 because the EMD is nonnegative. We need to show that $\mathrm{EMD}_{\mathcal{G}}(E_1, E_2) = 0$ iff $E_1 = E_2$. The nontrivial direction is to show that $\mathrm{EMD}_{\mathcal{G}}(E_1, E_2) = 0$ implies $E_1 = E_2$. Fix equivalence class representatives $\mathbf{x} \in E_1$ and $\mathbf{y} \in E_2$. If $\mathrm{EMD}_{\mathcal{G}}(E_1, E_2) = 0$, then by definitions (6.3) and (6.6) there exists $g \in \mathcal{G}$ such that $\mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) = 0$. Since the EMD is a metric, we must have $\mathbf{x} = g(\mathbf{y})$. It follows that $E_1 = E_2$. The symmetry of $\mathrm{EMD}_{\mathcal{G}}$ follows from the symmetry of the EMD. Finally, we need to show that the triangle inequality holds. Suppose E_1, E_2, and E_3 are equivalence classes with representative distributions $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$, and that

$$\begin{aligned} \mathrm{EMD}_{\mathcal{G}}(E_1, E_2) &= \mathrm{EMD}(g_x(\mathbf{x}), \mathbf{y}) \quad \text{and} && (6.7) \\ \mathrm{EMD}_{\mathcal{G}}(E_2, E_3) &= \mathrm{EMD}(\mathbf{y}, g_z(\mathbf{z})). && (6.8) \end{aligned}$$

Here g_x and g_z are the transformations of $\mathbf{x}$ and $\mathbf{z}$ that yield the minimum value in (6.3). Since $g_x(\mathbf{x}) \sim \mathbf{x}$ and $g_z(\mathbf{z}) \sim \mathbf{z}$, it follows from (6.3) and (6.6) that

$$\mathrm{EMD}_{\mathcal{G}}(E_1, E_3) \leq \mathrm{EMD}(g_x(\mathbf{x}), g_z(\mathbf{z})).$$

But the EMD is a metric between distributions, so it obeys the triangle inequality and

$$\mathrm{EMD}(g_x(\mathbf{x}), g_z(\mathbf{z})) \leq \mathrm{EMD}(g_x(\mathbf{x}), \mathbf{y}) + \mathrm{EMD}(\mathbf{y}, g_z(\mathbf{z})).$$

Combining the previous two inequalities with (6.7) and (6.8) gives the triangle inequality

$$\mathrm{EMD}_{\mathcal{G}}(E_1, E_3) \leq \mathrm{EMD}_{\mathcal{G}}(E_1, E_2) + \mathrm{EMD}_{\mathcal{G}}(E_2, E_3)$$

that we desire. ■

For simplicity, we write about $\mathrm{EMD}_{\mathcal{G}}(\mathbf{x}, \mathbf{y})$ in the remaining sections of this chapter as if it were just $\min_{g \in \mathcal{G}} \mathrm{EMD}(\mathbf{x}, g(\mathbf{y}))$.

6.2 A Direct Algorithm

The transformed distribution $g(\mathbf{y}) = (g(Y), u) \in \mathbf{D}^{K,n}$ has the same weights as the original distribution $\mathbf{y}$. Thus $\mathcal{F}(\mathbf{x}, \mathbf{y}) = \mathcal{F}(\mathbf{x}, g(\mathbf{y}))$ and

$$\mathrm{EMD}_{\mathcal{G}}(\mathbf{x}, \mathbf{y}) = \frac{\min_{g \in \mathcal{G}, F \in \mathcal{F}(\mathbf{x},\mathbf{y})} \mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y}))}{\min(w_\Sigma, u_\Sigma)}. \tag{6.9}$$

Clearly, it suffices to minimize the work $h(F, g) = \mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y}))$ over the region $R(\mathbf{x}, \mathbf{y}) = \{\ (F, g)\ :\ F \in \mathcal{F}(\mathbf{x}, \mathbf{y}), g \in \mathcal{G}\ \} = \mathcal{F}(\mathbf{x}, \mathbf{y}) \times \mathcal{G}$.

The function $h(F, g)$ is linear in F. It follows that for g fixed, the minimum value $\min_{F \in \mathcal{F}(\mathbf{x},\mathbf{y})} h(F, g)$ is achieved at one of the vertices (dependent on g) of the convex polytope $\mathcal{F}(\mathbf{x}, \mathbf{y})$. If we let $V(\mathbf{x}, \mathbf{y}) = \{\ v_1(\mathbf{x}, \mathbf{y}),\ \ldots,\ v_N(\mathbf{x}, \mathbf{y})\ \}$ denote this finite set of vertices, then

$$\min_{(F,g) \in R(\mathbf{x},\mathbf{y})} h(F, g) = \min_{k=1,\ldots,N} \min_{g \in \mathcal{G}} h(v_k(\mathbf{x}, \mathbf{y}), g). \tag{6.10}$$

Assuming that we can solve the innermost minimization problem on the right-hand side of (6.10), we can compute the numerator in (6.9) by simply looping over all the vertices in $V(\mathbf{x}, \mathbf{y})$. 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 transformation, it is not practical because N is usually very large even for relatively small values of m and n. The worst case complexity of the number of vertices in the feasible convex polytope for a linear program is exponential in the minimum of the number of variables and constraints.[1] The beauty of the simplex algorithm 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 almost always no larger in order than the maximum of the number of variables and the number of constraints ([58]). In the next section, we give an iterative algorithm that generates a sequence of (flow,transformation) pairs for which the amount of work decreases or remains constant at every step.

6.3 The FT Iteration

The work function $h(F, g)$ to minimize depends on both a flow vector F and a transformation g. Given either variable, we can solve for the optimal value of the other. This leads us to an iteration which alternates between finding the best flow for a given transformation, and

[1] For a balanced transportation problem with m suppliers and n demanders, there are mn variables and $m + n$ constraints (not including the nonnegativity constraints on the variables).

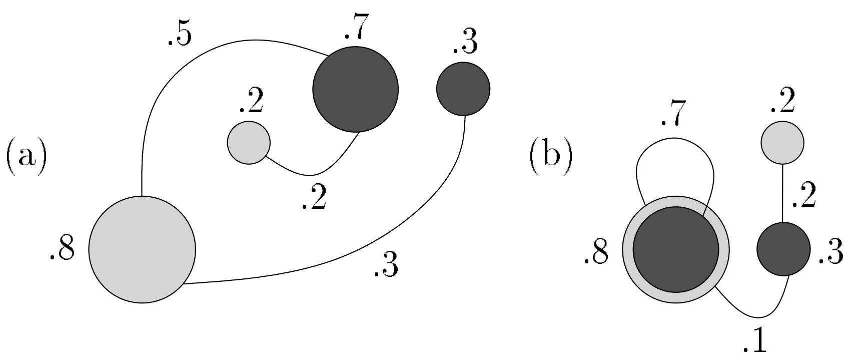

Figure 6.1: FT Iteration Example. See the text for an explanation.

the best transformation for a given flow. The flow step establishes correspondences that minimize the work for a fixed configuration of distribution points, while the transformation step moves the distribution points around so that the work is minimized for a given set of correspondences. By alternating these steps we obtain a sequence of (flow,transformation) pairs for which the amount of work decreases or remains constant at every step.

In this section, we consider distribution transformations that alter only the points of a distribution, leaving distribution weights unchanged. If g is such a transformation, then $\mathcal{F}(\mathbf{x}, g(\mathbf{y})) = \mathcal{F}(\mathbf{x}, \mathbf{y})$ since $\mathbf{y}$ and $g(\mathbf{y})$ have the same set of weights. Consider the following iteration that begins with an initial transformation $g^{(0)}$:

$$F^{(k)} = \arg\left(\min_{F\in\mathcal{F}(\mathbf{x},g^{(k)}(\mathbf{y}))=\mathcal{F}(\mathbf{x},\mathbf{y})} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\, d\left(x_i, g^{(k)}(y_j)\right)\right), \qquad (6.11)$$

$$g^{(k+1)} = \arg\left(\min_{g\in\mathcal{G}} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d(x_i, g(y_j))\right). \qquad (6.12)$$

The minimization problem on the right-hand side of (6.11) is the familar transportation problem. For now, we assume that there is an algorithm to solve for the optimal transformation in (6.12). This problem is the subject of section 6.4. Since this iteration alternates between finding an optimal Flow and an optimal Transformation, we refer to (6.11) and (6.12) as the *FT* iteration. It can be applied to equal-weight and unequal-weight distributions.

Figure 6.1(a) shows an example with a dark and a light distribution that we will match under translation starting with $g^{(0)} = 0$. The best flow $F^{(0)}$ for $g^{(0)}$ is shown by the labelled arcs connecting dark and light weights. This flow matches half (.5) the weight over a large distance. We should expect the best translation for $F^{(0)}$ to move the .7 dark weight closer to the .8 light weight in order to decrease the total amount of work done by $F^{(0)}$. Indeed, $g^{(1)}$ aligns these two weights as shown in Figure 6.1(b). The best flow $F^{(1)}$ for this translation matches all of the .7 dark weight to the .8 light weight. No further translation improves the work – $g^{(2)} = g^{(1)}$ and the FT iteration converges.

The flow and transformation iterates $F^{(k)}$ and $g^{(k)}$ define the WORK and EMD iterates

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

The order of evaluation is

$$\underbrace{g^{(0)} \Leftrightarrow\!\!\rightarrow F^{(0)}}_{\mathrm{WORK}^{(0)},\,\mathrm{EMD}^{(0)}} \Leftrightarrow\!\!\rightarrow \underbrace{g^{(1)} \Leftrightarrow\!\!\rightarrow F^{(1)}}_{\mathrm{WORK}^{(1)},\,\mathrm{EMD}^{(1)}} \Leftrightarrow\!\!\rightarrow \cdots .$$

By (6.11), we have

$$\begin{aligned} \mathrm{WORK}^{(k+1)} &= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k+1)} d\left(x_i, g^{(k+1)}(y_j)\right) \\ &\leq \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, g^{(k+1)}(y_j)\right). \end{aligned} \tag{6.13}$$

In detail, $F^{(k)} \in \mathcal{F}(\mathbf{x},\mathbf{y}) = \mathcal{F}(\mathbf{x}, g^{(k)}(\mathbf{y}))$, while $F^{(k+1)} \in \mathcal{F}(\mathbf{x},\mathbf{y}) = \mathcal{F}(\mathbf{x}, g^{(k+1)}(\mathbf{y}))$ is optimal for $g^{(k+1)}$ over all flows in $\mathcal{F}(\mathbf{x},\mathbf{y})$. Therefore using $F^{(k+1)}$ with $g^{(k+1)}$ results in less work than using $F^{(k)}$ with $g^{(k+1)}$. From (6.12), we know

$$\begin{aligned} \mathrm{WORK}^{(k)} &= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, g^{(k)}(y_j)\right) \\ &\geq \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^{(k)} d\left(x_i, g^{(k+1)}(y_j)\right). \end{aligned} \tag{6.14}$$

Combining (6.13) and (6.14),

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

The decreasing sequence $\mathrm{WORK}^{(k)}$ is bounded below by zero, and hence it converges ([38], pp. 49–50). There is, however, no guarantee that the work iteration converges to the global minimum of $h(F, g) = \mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y}))$.

Using the exact same iteration with $\mathcal{F}^{\gamma}(\mathbf{x},\mathbf{y})$ in place of $\mathcal{F}(\mathbf{x},\mathbf{y})$ will also yield a decreasing sequence of WORK values (and, hence, a decreasing sequence of EMD values). This is because $\mathcal{F}^{\gamma}(\mathbf{x},g(\mathbf{y})) = \mathcal{F}^{\gamma}(\mathbf{x},\mathbf{y})$ when g does not change distribution weights. Therefore, the FT iteration can also be used in an attempt to compute the partial EMD under transformation when the transformations do not change the weights of a distribution. Please refer back to section 4.4.1 for the details of the partial EMD.

6.3.1 Similar Work

The FT iteration is similar to the ICP (Iterative Closest Point) algorithm ([5]) used to register 3D shapes. The computation of the optimal flow between distributions in the FT iteration plays the role of the computation of closest "model shape" points to the "data shape" points in the ICP iteration. Both these steps determine correspondences used to compute a transformation that improves the EMD/registration. There are, however, a number of important differences between the two algorithms and the contexts in which they are applied.

As we noted in section 4.3.1, the EMD provides a distance between point sets (which are distributions in which all weights are equal to one) as well as general distributions. The ICP algorithm is used to register a data shape defined as a point set with a model shape defined by a set of geometric primitives such as points, line segments, curves, etc.. If the model shape is also defined as a set of points, then the ICP algorithm also seeks to minimize the distance between two point sets under a transformation. The notion of point set distance defined by the EMD, however, is different than the notion of distance used in the ICP formulation. The ICP algorithm's correspondence step computes the nearest neighbor in the model shape for each data shape point, and sums up all these distances. The same model shape point may be the nearest neighbor of many data shape points in the distance sum computation. Therefore, the ICP algorithm uses a Hausdorff-like distance between point sets. In contrast, when $\text{EMD}_{\mathcal{G}}$ is used to match point sets under a transformation, the constraints that define the EMD imply a one-to-one matching of the points (see section 4.3.1). The correspondence step for the FT iteration requires the solution of an assignment problem, whereas the correspondence step in the ICP algorithm matches each data shape point independently to its closest model shape point. The unconstrained matching in the ICP algorithm will obviously be faster to compute than the constrained matching specified by the EMD, but the two iterations are trying to minimize different point set distance metrics.

Using our EMD framework instead of the ICP framework to match point sets under a transformation has the advantage that it can find the best subsets (with size specified by γ) of the given sets to match. In fact, the partial match parameter γ can be used to align different sub-distributions (subsets) of two distributions (point sets) using different transformations. For example, one could find the transformation that works well for $\gamma =$ 20% of the data, remove the matched data, and repeat. The ICP algorithm could also be applied in this piecewise manner, but the user must select the subsets of the data shape to be matched, not just the subset size as in the partial EMD case.

Limiting our comparison of the FT iteration and the ICP iteration to point sets is unfair to the FT iteration since the EMD can be used to match distributions of mass which are more general than point sets. The mass at a point in one distribution can be matched to the mass at many points in the other distribution, and vice-versa. The FT iteration in general provides a many-to-many matching of distribution points, while the ICP iteration (applied to a model shape point set) gives a many-to-one matching of model shape points to data shape points, and the FT iteration applied to point sets gives a one-to-one matching of points in the two sets. Furthermore, the amount of mass matched between two points is used to weight the distance between the points; all the point distances have weight one in the ICP iteration and the FT iteration applied to point sets. The many-to-many matching specified by the EMD is constrained by the distribution masses in such a way that the matching process represents a morphing of one distribution into the another. As we shall see in section 6.5, the FT iteration as presented thus far can be modified to handle the case in which transformations are allowed to modify both the distribution points and their corresponding masses.

Another well-known application of the alternation idea is the *Expectation-Maximization* or EM algorithm ([47, 48]) for computing mixture models in statistics. In this problem, observed data are assumed to arise from some number of parametrized distributions. The goal is to determine which data come from which distributions and to compute the parameter values of the distributions. The EM algorithm alternates between finding the expected assignments[2] of the data to the distributions with fixed defining parameters (the E-step), and finding the maximum likelihood estimate of the parameter values given the expected assignments (the M-step). The function to be optimized in this case is a likelihood function, and the optimization problem is a maximization.

[2] The E-step computes the expected value $E[Z_{ij}] = P(Z_{ij} = 1)$, where $Z_{ij} = 1$ if the jth observation arises from the ith distribution, and $Z_{ij} = 0$ otherwise.

6.3.2 Convergence Properties

One way for the WORK iteration to converge is if $F^{(k)}$ is returned in step (6.11) as an optimal flow for $g^{(k)}$, and $g^{(k+1)} = g^{(k)}$ is returned in step (6.12) as an optimal transformation for $F^{(k)}$. Denote the indicator function for this event as $\mathrm{MUTUAL}\left(F^{(k)}, g^{(k)}\right)$, since $F^{(k)}$ is optimal for $g^{(k)}$, and $g^{(k)}$ is optimal for $F^{(k)}$. It is clear that

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

The WORK iteration converges to either a local minimum or a saddle point value if $\mathrm{MUTUAL}\left(F^{(k)}, g^{(k)}\right)$ is true.[3]

Now suppose that the routine that solves the linear program (LP) in (6.11) always returns a vertex of $\mathcal{F}(\mathbf{x},\mathbf{y})$. The simplex algorithm and the transportation simplex algorithm, for example, always return 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 $\mathcal{F}(\mathbf{x},\mathbf{y})$, 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 transformation iterate (other than the initial transformation) must be an optimal transformation 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

$$\mathrm{WORK}^{(k)} = \mathrm{WORK}^{(k+1)} = \cdots\,. \tag{6.15}$$

Since there are only a finite number of pairs (F,t) visited, condition (6.15) implies that there must be a repeating cycle of pairs:

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

[3] If $g^{(k)}$ occurs in the interior of $R(F^{(k)}) = \left\{\, g \,:\, F^{(k)} \in \arg\min_{F\in\mathcal{F}} \mathrm{WORK}(F,x,g(y)) \,\right\}$, then $(F^{(k)}, g^{(k)})$ cannot be a saddle point and the WORK iteration converges to a local minimum of $\mathrm{WORK}(F,\mathbf{x},g(\mathbf{y}))$. The argument is toward the end of section 6.7.4, the paragraph beginning "Let us now explicitly connect ...". In general, the possibility convergence to a saddle point cannot be eliminated. All we know is that along the F axis though $g^{(k)}$, the minimum occurs at $F^{(k)}$, and along the g axis though $F^{(k)}$, the minimum occurs at $g^{(k)}$. This does not imply anything about the value of WORK close to $(F^{(k)}, g^{(k)})$ along a "diagonal" through $(F^{(k)}, g^{(k)})$.

For $r > 1$, the WORK iteration converges even though the flow and transformation iterations do not converge. However, such a nontrivial (flow,transformation) cycle is unstable in the sense that it can be broken (for any real problem data) by perturbing one of the transformation 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 $\left(F^{(k)}, g^{(k)}\right)$ is exactly the condition MUTUAL$\left(F^{(k)}, g^{(k)}\right)$, and we previously argued that the WORK iteration converges to a critical value in this case.

Finally, let us note that the WORK sequence will stabilize at the global minimum once $F^{(k)} = F^*$, where (F^*, g^*) is optimal for some g^*. This is because $g^{(k+1)}$ and g^* both solve (6.12), so $h(F^{(k)}, g^{(k+1)}) = h(F^*, g^*)$, which is the global minimum of the WORK function. Since h can only decrease or remain the same with successive flow and transformation iterates, and h can never be less than the global minimum of the WORK function, we must have

$$h(F^*, g^*) = \mathrm{WORK}^{(k+1)} = h(F^{(k+1)}, g^{(k+1)}) = \mathrm{WORK}^{(k+2)} = h(F^{(k+2)}, g^{(k+2)}) = \cdots .$$

Similarly, if the transformation iteration ever reaches a transformation $g^{(k)} = g^*$ at which the minimum value of WORK occurs with some flow F^*, then the WORK iteration converges to the global minimum. Here we need the fact that $F^{(k)}$ and F^* both solve (6.11).

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 $\mathcal{F}(\mathbf{x}, \mathbf{y})$. In this case, the (flow,transformation) iterates must cycle. A cycle of length $r > 1$ will almost never occur, and a cycle of length $r = 1$ implies that the (flow,transformation) 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 transformation, then the WORK iteration converges to the global minimum. Global convergence will also occur if the transformation iteration ever reaches a transformation at which the minimum WORK occurs with a suitable choice of flow.

Although we do not explore the possibility here, perhaps convergence of the FT iteration can be accelerated by using the EMD values at the past few transformation iterates to predict the transformation at which the EMD will be minimized. If the predicted transformation causes an increase in the EMD, we could discard the prediction and just use the solution to (6.12) as usual to define the next transformation iterate. This approach successfully accelerated the convergence of the previously mentioned ICP iteration ([5]).

6.4 The Optimal Transformation Problem

Now consider the problem (6.12) of solving for the optimal transformation for a fixed $F = (f_{ij})$:

$$\min_{g \in \mathcal{G}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(x_i, g(y_j)). \tag{6.16}$$

If we let

$$\begin{aligned} [a_1 \cdots a_N] &= [x_1 \cdots x_1 x_2 \cdots x_2 \cdots x_m \cdots x_m], && (6.17) \\ [b_1 \cdots b_N] &= [y_1 \cdots y_n y_1 \cdots y_n \cdots y_1 \cdots y_n], \quad \text{and} && (6.18) \\ [c_1 \cdots c_N] &= [f_{11} \cdots f_{1n} f_{21} \cdots f_{2n} \cdots f_{m1} \cdots f_{mn}], && (6.19) \end{aligned}$$

where $N = mn$, then the *optimal transformation problem* (6.16) can be rewritten as

$$\min_{g \in \mathcal{G}} \sum_{k=1}^{N} c_k d(a_k, g(b_k)). \tag{6.20}$$

In this form, the optimal transformation problem can be stated as follows: given a weighted correspondence between point sets, find a transformation of the points in one set that minimizes the weighted sum of distances to corresponding points in the other set.[4] We now discuss the solution of the optimal transformation problem for translation, Euclidean, similarity, linear, and affine transformations with d equal to the L_2-distance squared, as well as the optimal translation problem with the L_2-distance, the L_1-distance, and a cyclic L_1-distance (to be explained shortly).

6.4.1 Translation

Suppose that $\mathcal{G} = \mathcal{T}$, the group of translations. If

$$d(x_i, y_j + t) = d(x_i \Leftrightarrow y_j, t), \tag{6.21}$$

then (6.20) can be written as

$$\min_{t \in \mathbf{R}^K} \sum_{k=1}^{N} c_k d(a_k \Leftrightarrow b_k, t).$$

[4] Note that some structure is lost in the rewrite from (6.16) to (6.20). In the setting of the FT iteration, the N points a_k consist of n copies of the set $\{x_i\}_{i=1}^{m}$, and the N points b_k consist of m copies of the set $\{y_j\}_{j=1}^{n}$.

Note that condition (6.21) holds for any L_p distance function d, as well as $d = L_2^2$. This minimization problem asks for a point t which minimizes a sum of weighted distances to a given set of points. We show how to solve this *minisum* problem when d is the L_2-distance squared, the L_1-distance, and the L_2-distance in sections 6.4.1.1, 6.4.1.2, and 6.4.1.3, respectively.

In section 6.4.1.4, we solve the optimal translation problem when points are located on a circle, and ground distance is the length of the shorter arc connecting two points. This problem arises, for example, when applying the FT iteration to compute an orientation-invariant EMD between texture signatures in log-polar frequency space ([69, 68]). The polar coordinates for spatial frequency (f_x, f_y) are (s, θ), where the scale $s = \sqrt{f_x^2 + f_y^2}$ and the orientation $\theta = \arctan(f_y, f_x)$. Roughly speaking, distributions are of the form $\mathbf{x} = \{((\widehat{s}_i, \theta_i), w_i)\}_{i=1}^m$, where $\widehat{s}_i = \log s_i$ and the ith element indicates that the texture has a fraction w_i of its total energy at scale s_i and orientation θ_i.

If we denote the distribution for another texture as $\{((\widehat{\alpha}_j, \psi_j), u_j)\}_{j=1}^n$, then the texture distance $\min_{\Delta\psi} \text{EMD}(\mathbf{x}, \{((\widehat{\alpha}_j, \psi_j + \Delta\psi), u_j)\})$ is invariant to texture orientation. The optimal translation problem is

$$\begin{aligned} &\min_{\Delta\psi} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} (|\widehat{s}_i - \widehat{\alpha}_j| + |\theta_i - (\psi_j + \Delta\psi)|_{2\pi}) = \\ &\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |\widehat{s}_i - \widehat{\alpha}_j| + \min_{\Delta\psi} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |\theta_i - (\psi_j + \Delta\psi)|_{2\pi}, \end{aligned} \tag{6.22}$$

where the absolute value subscript indicates distances are measured modulo 2π. The minimization problem on the right-hand side of (6.22) is the subject of section 6.4.1.4.

Notice that using the logarithm of the scale in the texture signatures implies that a scale change by factor k results in a shift by $\widehat{k} = \log k$ along the log-scale axis. A scale-invariant texture distance measure is $\min_{\widehat{k}} \text{EMD}(\mathbf{x}, \{((\widehat{\alpha}_j + \widehat{k}, \psi_j), u_j)\})$. The optimal translation problem in this case is

$$\begin{aligned} &\min_{\widehat{k}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} (|\widehat{s}_i - (\widehat{\alpha}_j + \widehat{k})| + |\theta_i - \psi_j|_{2\pi}) = \\ &\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |\theta_i - \psi_j|_{2\pi} + \min_{\widehat{k}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |\widehat{s}_i - (\widehat{\alpha}_j + \widehat{k})|. \end{aligned} \tag{6.23}$$

The L_1 minimization problem on the right-hand side of (6.23) is solved in section 6.4.1.2.

Finally, the texture distance $\min_{\widehat{k}, \Delta\psi} \text{EMD}(\mathbf{x}, \{((\widehat{\alpha}_j + \widehat{k}, \psi_j + \Delta\psi), u_j)\})$ is invariant to both texture scale and orientation. The associated optimal translation problem

$$\min_{\widehat{k},\Delta\psi} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}(|\widehat{s}_i - (\widehat{\alpha}_j + \widehat{k})| + |\theta_i - (\psi_j + \Delta\psi)|_{2\pi}) = \tag{6.24}$$
$$\min_{\widehat{k}} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}|\widehat{s}_i - (\widehat{\alpha}_j + \widehat{k})| + \min_{\Delta\psi} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}|\theta_i - (\psi_j + \Delta\psi)|_{2\pi}$$

can be solved by solving the minimization problems on the right-hand side of (6.24) separately. In general, of course, the separability of the L_1 distance into a sum of component distances means that we can solve the optimal translation problem under L_1 in any number of dimensions, where each dimension may have a different wrap around period or no wrap around at all.

6.4.1.1 Minimizing a Weighted Sum of Squared L_2 Distances

If d is the L_2-distance squared, then it is well-known that the unique optimal translation is given by the centroid difference

$$t^*_{L_2^2} = \overline{a} - \overline{b} = \frac{\sum_{k=1}^{N} c_k a_k}{c_\Sigma} - \frac{\sum_{k=1}^{N} c_k b_k}{c_\Sigma}.$$

This result is easily proven using standard calculus.

6.4.1.2 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

$$\begin{aligned}
\min_p \sum_{i=1}^{n} w_i ||p - p_i||_1 &= \min_p \sum_{i=1}^{n} w_i \sum_{k=1}^{K} \left| p^{(k)} - p_i^{(k)} \right| \\
&= \min_p \sum_{k=1}^{K} \left(\sum_{i=1}^{n} w_i \left| p^{(k)} - p_i^{(k)} \right| \right) \\
\min_p \sum_{i=1}^{n} w_i ||p - p_i||_1 &= \sum_{k=1}^{K} \left(\min_{p^{(k)}} \sum_{i=1}^{n} w_i \left| p^{(k)} - p_i^{(k)} \right| \right),
\end{aligned} \tag{6.25}$$

where $p^{(k)}$ and $p_i^{(k)}$ are the kth components of p and p_i, respectively.[5] Thus, a solution to the problem in one dimension gives a solution to the problem in K dimensions by simply collecting the optimal location for each of the one-dimensional problems into a K-dimensional vector. We shall see that an optimal location for the one-dimensional problem in dimension k is the (weighted) median of the values $p_1^{(k)}, \ldots, p_n^{(k)}$. A point p at which the minimum (6.25) is achieved is thus called a *coordinate-wise median* of $p_1, \ldots, p_n$ (with weights $w_1, \ldots, w_n$).

Now suppose $p_1 \le p_2 \le \cdots \le p_n$ 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

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

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

$$m_l = \sum_{i=1}^{l} w_i - \sum_{i=l+1}^{n} w_i \tag{6.26}$$

denote the slope of $g(p)$ over $[p_l, p_{l+1}]$, then $-w_\Sigma = m_0 < m_1 < \cdots < m_n = w_\Sigma$ (assuming $w_i > 0 \ \forall i$), and $m_{l+1} = m_l + 2w_{l+1}$. 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

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

If $m_{l^*} \ne 0$, 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 6.2. In the special case of 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 (6.26) 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 $p_{\lceil n/2 \rceil}$.

[5] In this section and the next, weights are denoted by w_i instead of c_k, the total number of points is denoted by n instead of N, and the summation of weighted distances is over the variable i instead of k. The points p_i are the differences $a_k - b_k$.

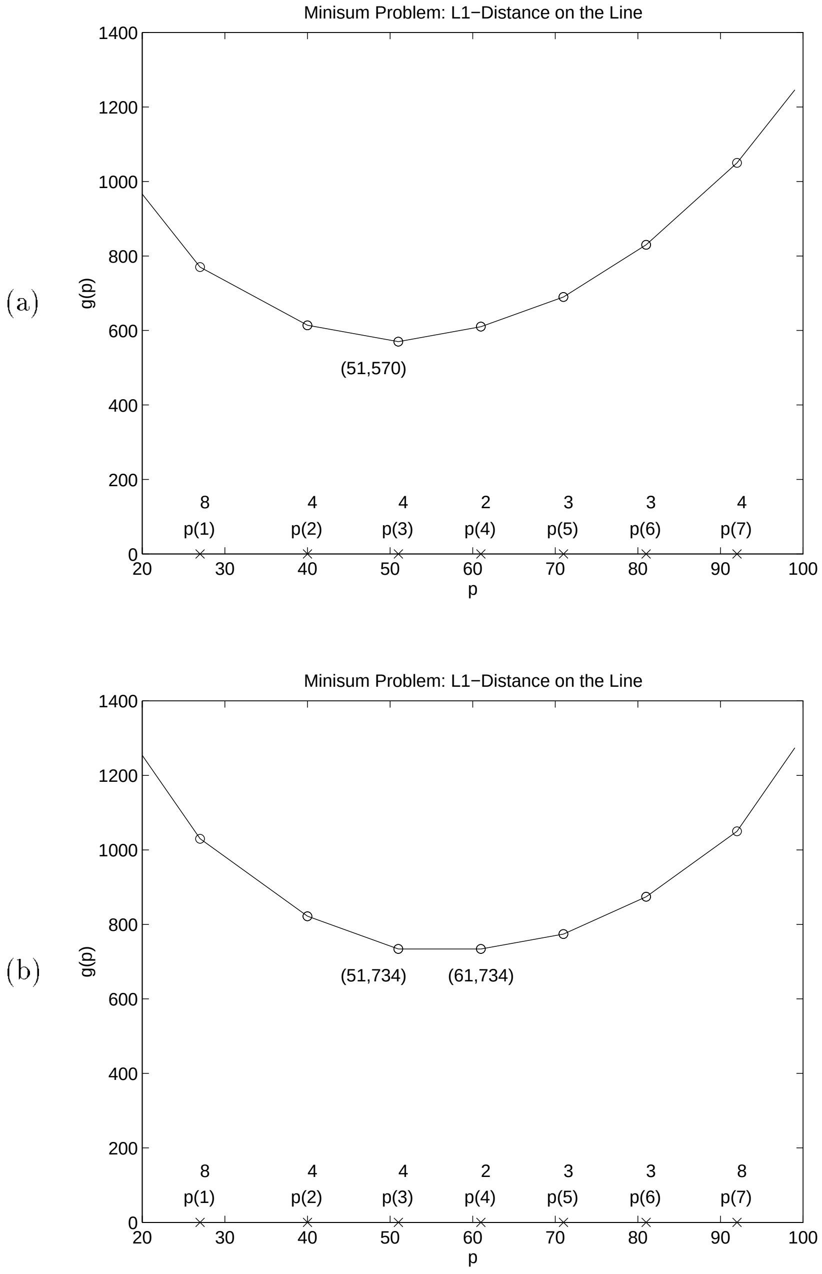

Figure 6.2: The Minisum Problem on the Line with Unequal Weights. (a) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [8, 4, 4, 2, 3, 3, 4]$: $l^* = 3$, $m_{l^*} > 0$, and there is a unique minimum at $p_3 = 51$. (b) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [8, 4, 4, 2, 3, 3, 8]$: $l^* = 3$, $m_{l^*} = 0$, and the minimum occurs at every value in $[p_3, p_4] = [51, 61]$.

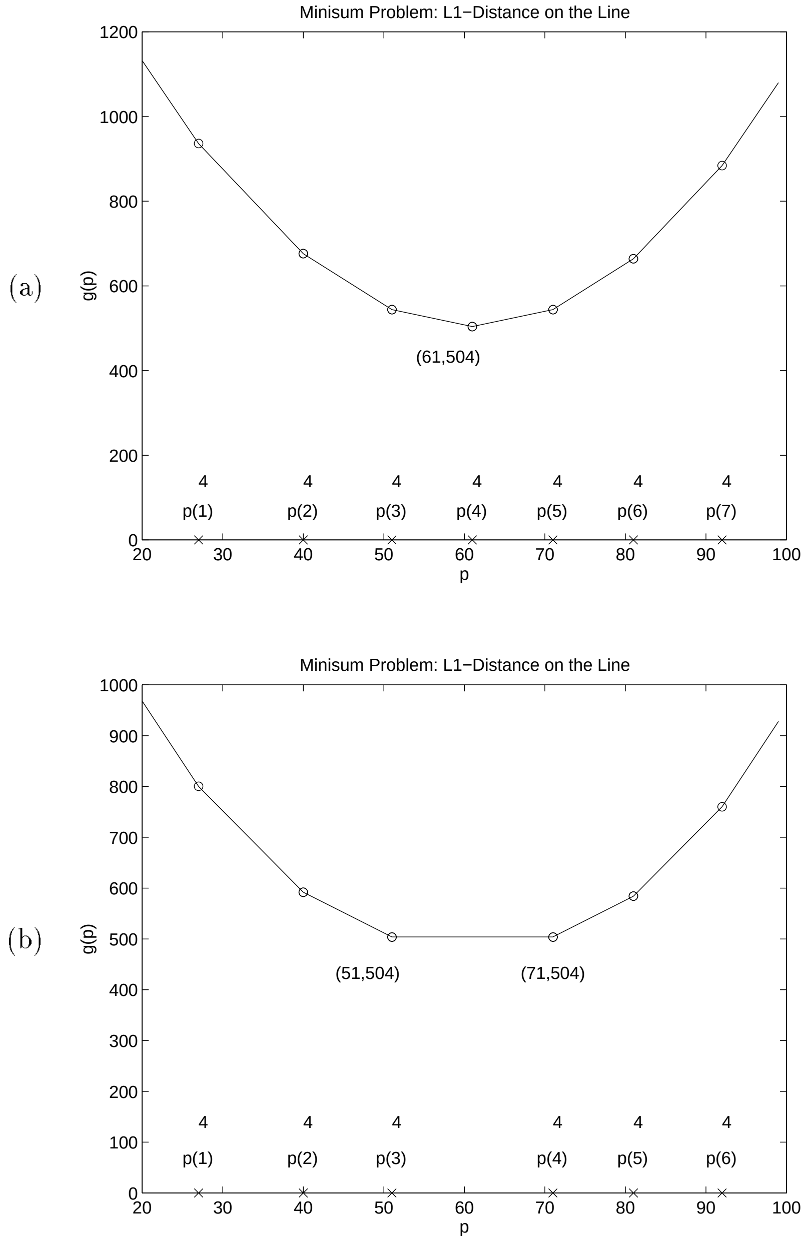

Figure 6.3: The Minisum Problem on the Line with Equal Weights. (a) $p = [27, 40, 51, 61, 71, 81, 92]$, $w = [4, 4, 4, 4, 4, 4, 4]$: $l^* = 4$, $m_{l^*} > 0$, and there is a unique minimum at the ordinary median $p_4 = 61$. (b) $p = [27, 40, 51, 71, 81, 92]$, $w = [4, 4, 4, 4, 4, 4]$: $l^* = 3$, $m_{l^*} = 0$, and the minimum occurs at every value in the interval $[p_3, p_4] = [51, 71]$.

See Figure 6.3(a). 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 6.3(b).

6.4.1.3 Minimizing a Weighted Sum of L_2 Distances

The next minisum problem that we consider is when d is the L_2-distance function:

$$\min_p \sum_{i=1}^{n} w_i ||p - p_i||_2 \tag{6.27}$$

A point p at which this minimum is achieved is called a *spatial median* of the points $p_1, \ldots, p_n$ (with weights $w_1, \ldots, w_n$). The minimization problem (6.27) has a long history ([87]), and has been referred to by many names, including the *Weber problem*, the *Fermat problem*, the *minisum problem*, and the *spatial median problem.* In [87], Wesolowsky suggests the *Euclidean Minisum Problem.*

A basic iteration procedure that solves (6.27) was proposed in 1937 by Weiszfeld ([84]). 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 the line, thereby decreasing its distance to all the given points, to obtain a better point). In this case, the algorithm given in section 6.4.1.2 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:

$$\frac{\partial g}{\partial p}(p) = \sum_{i=1}^{n} \frac{w_i (p - p_i)}{||p - p_i||_2} \qquad \text{if } p^{(k)} \neq p_1, \ldots, p_n.$$

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. ([42])
- If no iterate $p^{(k)}$ is equal to one of the given points, then the iteration converges to the global minimum location of $g(p)$. ([42])
- 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. ([7])
- If the optimal location is *not* at one of the given points, then convergence will be linear. ([41])
- If the optimal location *is* at one of the given points, then convergence can be linear, superlinear, or sublinear. ([41])

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 [7] 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 [18] for an accelerated Weiszfeld procedure.

6.4.1.4 Minimizing a Weighted Sum of Cyclic L_1 Distances

In this section, we study the optimal translation problem on the real line when the feature domain is circular. In other words, we assume the feature points are real numbers which are defined only modulo T. Also, we assume the ground distance is the *cyclic L_1-distance*

$$d_{L_1,T}(x,y) = \min_{k \in \mathbf{Z}} |(x+kT) - y|.$$

If we identify feature values with arclengths on a circle of perimeter T, then $d_{L_1,T}(x,y)$ measures the smaller of the two arclengths that connect x to y along the circle. It is easy to prove that $0 \leq d_{L_1,T}(x,y) \leq T/2$. The intuition here is that a point should never have to travel more than half the circle to arrive at another point. Suppose, for example, that features are angles in radians ($T = 2\pi$), $x = \pi/4 = 45°$, and $y = 11\pi/6 = 330°$. Then $d_{L_1,T}(x,y) = d_{L_1,T}(\pi/4, 11\pi/6) = 5\pi/12 = 75°$, where the minimum is achieved at $k = 1$. It should also be clear that $d_{L_1,T}$ is cyclic with period T in both arguments: $d_{L_1,T}(x+T,y) = d_{L_1,T}(x,y+T) = d_{L_1,T}(x,y)$.

In order to apply the FT iteration for translation with the cyclic L_1-distance, we need to minimize

$$\text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus t) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{L_1,T}(x_i, y_j + t).$$

over t. As is the L_1 case in section 6.4.1.2, the multidimensional case in $\mathbf{R}^K$ can be solved by solving K one-dimensional problems.[6] Therefore, consider the minimization problem

$$\min_{t\in\mathbf{R}} \text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus t) = \min_{t\in\mathbf{R}} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \min_{k\in\mathbf{Z}} |(x_i + kT) - (y_j + t)| \tag{6.28}$$

given a fixed flow F. Since $d_{L_1,T}$ is cyclic with period T, the WORK function is cyclic in t with period T: $\text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus (t+T)) = \text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus t)$. Therefore, for every feasible flow F there will be a WORK minimizing translation $t \in [0, T)$. We can also assume that $x_i, y_j \in [0, T)$ since these numbers need only be defined up to a multiple of T when using ground distance $d_{L_1,T}$.

The inner minimization of (6.28) can be trivially rewritten as

$$\min_{k\in\mathbf{Z}} |(x_i + kT) - (y_j + t)| = \min_{k\in\mathbf{Z}} |kT - (y_j + t - x_i)|.$$

If we restrict $x_i, y_j, t \in [0, T)$, then $(y_j + t - x_i) \in (-T, 2T)$. The above minimum will never be achieved outside the set $\{ -1, 0, 1, 2 \}$, so

$$\min_{k\in\mathbf{Z}} |(x_i + kT) - (y_j + t)| = \min_{k\in\{ -1,0,1,2 \}} |(x_i + kT) - (y_j + t)| \quad \text{for } x_i, y_j, t \in [0, T).$$

If we let

$$h(t) = h(F, \mathbf{x}, \mathbf{y}, t) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \min_{k\in\{ -1,0,1,2 \}} |kT - (y_j + t - x_i)|, \tag{6.29}$$

then we have argued that

$$\min_{t\in\mathbf{R}} \text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus t) = \min_{t\in[0,T)} h(t) \qquad \text{if } x_i, y_j \in [0, T).$$

We now consider the function $h(t)$ in more detail.

[6] The period can be different in each dimension.

In order to better understand $h(t)$, we can partition the real line into intervals over which $\arg\min_{k\in\{-1,0,1,2\}} |kT - (y_j + t - x_i)|$ is constant. If we let $z_{ij} = x_i - y_j$, then $|kT - (y_j + t - x_i)| = |kT + z_{ij} - t|$, and

$$\arg \min_{k\in\{-1,0,1,2\}} |kT + z_{ij} - t| = \begin{cases} -1 & \text{if } -\infty \le t < -\frac{1}{2}T + z_{ij} \\ 0 & \text{if } -\frac{1}{2}T + z_{ij} \le t < \frac{1}{2}T + z_{ij} \\ 1 & \text{if } \frac{1}{2}T + z_{ij} \le t < \frac{3}{2}T + z_{ij} \\ 2 & \text{if } \frac{3}{2}T + z_{ij} \le t < +\infty \end{cases} .$$

The plan now is to divide each of the above intervals into two parts: one in which $kT+z_{ij} < t$, and the other in which $kT+z_{ij} \ge t$. This will allow us to express $\min_{k\in\{-1,0,1,2\}} |kT+z_{ij}-t|$ as a linear function in t over these subintervals (i.e. we can eliminate the min operator and the absolute value function). Toward this end, define the intervals

$$\begin{aligned} I_{ijkl} &= [(k+\tfrac{l}{2})T + z_{ij}, (k+\tfrac{l+1}{2})T + z_{ij}) && i=1,\ldots,m, j=1,\ldots,n, \\ & && k=-1,0,1,2, l=-1,0, \\ & && (k,l) \ne (-1,-1), (k,l) \ne (2,0), \\ I_{ij(-1)(-1)} &= [-\infty, -T + z_{ij}) && i=1,\ldots,m, j=1,\ldots,n, \quad \text{and} \\ I_{ij20} &= [2T + z_{ij}, +\infty) && i=1,\ldots,m, j=1,\ldots,n. \end{aligned}$$

Here

$$\begin{aligned} I_{ij(-1)(-1)} \cup I_{ij(-1)0} &= [-\infty, -\frac{1}{2}T + z_{ij}], \\ I_{ijk(-1)} \cup I_{ijk0} &= [(k-\frac{1}{2})T + z_{ij}, (k+\frac{1}{2})T + z_{ij}) \quad k=0,1, \quad \text{and} \\ I_{ij2(-1)} \cup I_{ij20} &= [\frac{3}{2}T + z_{ij}, +\infty), \end{aligned}$$

so that $\arg\min_{k\in\{-1,0,1,2\}} |kT + z_{ij} - t| = k^*$ for $t \in I_{ij(k^*)(-1)} \cup I_{ij(k^*)0}$. Furthermore,

$$\min_{k\in\{-1,0,1,2\}} |kT + z_{ij} - t| = \begin{cases} k^*T + z_{ij} - t & \text{if } t \in I_{ij(k^*)(-1)} \\ t - k^*T - z_{ij} & \text{if } t \in I_{ij(k^*)0} \end{cases} .$$

The above notation will now be used to rewrite $h(t)$ in a form that makes its structure more apparent.

We can get rid of the absolute value and minimization in (6.29) with the following algebraic manipulations:

$$\begin{aligned}
h(t) &= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \min_{k\in\{\,-1,0,1,2\,\}} |kT + z_{ij} - t| \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \sum_{k=-1}^{2}\sum_{l=-1}^{0} [t \in I_{ijkl}]|kT + z_{ij} - t| \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \sum_{k=-1}^{2} ([t \in I_{ijk(-1)}](kT + z_{ij} - t) + [t \in I_{ijk0}](t - kT - z_{ij})) \\
h(t) &= \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \sum_{k=-1}^{2} ([t \in I_{ijk0}] - [t \in I_{ijk(-1)}])\right) t + \qquad (6.30) \\
&\quad \left(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \sum_{k=-1}^{2} ([t \in I_{ijk(-1)}] - [t \in I_{ijk0}])(kT + z_{ij})\right).
\end{aligned}$$

If we let $e_{ijkl} = (k + \frac{l}{2})T + z_{ij}$, then the breakpoint set

$$E = \{\ e_{ijkl}\ :\ i \in [1..m], j \in [1..n], k \in [-1..2], l \in [-1..0], (k,l) \neq (-1,-1)\ \}$$

divides the real line into intervals over which the coefficient of t and the coefficient of $1 = t^0$ in (6.30) are constant. On each such interval, the equation of $h(t)$ is that of a line. Therefore, $h(t)$ is a piecewise linear function of t.

The minimum of $h(t)$ over $[0, T)$ can be computed by visiting the breakpoints e_{ijkl} in sorted order, updating the line equation for $h(t)$ as we go along. The line function for $h(t)$ at $t = -\infty$ is given by

$$h(-\infty) = \left(-\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\right) t + \left(-T\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}z_{ij}\right).$$

This follows from (6.30) and the fact that $-\infty \in I_{ij(-1)(-1)}$. Thus the sweep algorithm sets the initial slope m to m_0 and the initial intercept b to b_0, where

$$m_0 = -\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \qquad \text{and} \qquad b_0 = -T\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}z_{ij}.$$

There are two types of elementary steps over a breakpoint $t = e_{ijkl}$. In the case $l = -1$, the sweep line moves from $I_{ij(k-1)0}$ into $I_{ijk(-1)}$. By subtracting the $I_{ij(k-1)0}$ terms and adding in the $I_{ijk(-1)}$ terms in (6.30), we see that the updates to the slope and intercept when $l = -1$ are

$$\begin{aligned} m &\leftarrow m - f_{ij} + (-f_{ij}) = m - 2f_{ij} \quad \text{and} && (6.31) \\ b &\leftarrow b - (-f_{ij}((k-1)T + z_{ij})) + f_{ij}(kT + z_{ij}) \\ &= b + f_{ij}((2k-1)T + 2z_{ij}). && (6.32) \end{aligned}$$

In the case $l = 0$, the sweep line moves from $I_{ijk(-1)}$ into I_{ijk0}. From (6.30), we see that the updates to the slope and intercept when $l = 0$ are

$$\begin{aligned} m &\leftarrow m - (-f_{ij}) + f_{ij} = m + 2f_{ij} \quad \text{and} && (6.33) \\ b &\leftarrow b - f_{ij}(kT + z_{ij}) + (-f_{ij}(kT + z_{ij})) \\ &= b - 2f_{ij}(kT + z_{ij}). && (6.34) \end{aligned}$$

The sweep algorithm maintains the minimum value seen so far as it proceeds from $t = -\infty$ to $t = \infty$. The value of the function $h(t)$ is checked at any breakpoint $0 \leq t = e_{ijkl} < T$ at which the slope of the line equation for $h(t)$ changes from negative to positive. The locations of such sign changes in slope are local minimum locations for $h(t)$ in $[0, T)$. Computing h at a local minimum location $t = e_{ijkl}$ is done via $h(e_{ijkl}) = me_{ijkl} + b$, where m and b are the slope and intercept *after* the update for passing e_{ijkl}. Since we want to compute the minimum of $h(t)$ over $t \in [0, T)$, we must also check the value of $h(0)$ when we have the formula for $h(t)$ over the interval that contains zero. Finally, we can stop the sweep once we reach a breakpoint $e_{ijkl} \geq T$.

One final note to make is that at most $m + n - 1$ of the mn values f_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, are nonzero if $F = (f_{ij})$ is an optimal vertex flow, as is returned by the transportation simplex algorithm ([32]). There is no reason to stop the sweep at e_{ijkl} for which $f_{ij} = 0$ since the values of m and b do not change at these points (see the update formulae (6.31)–(6.34)). This is obvious since the summation for $h(t)$ in (6.29) is the same with or without the (i, j)th term when $f_{ij} = 0$. The desired minimum can be computed by sweeping over the set

$$E' = \{\ e_{ijkl}\ :\ f_{ij} \neq 0, i \in [1..m], j \in [1..n], k \in [-1..2], l \in [-1..0], (k, l) \neq (-1, -1)\ \}$$

instead of the set E. Note that $|E| = 7mn$, while $|E'| \leq 7(m + n - 1)$. The sorting of the points in E' takes time $O((m + n)\log(m + n))$, and then the sweep over the points in E' takes time $O(m + n)$ since only a constant amount of work needs to be done at each elementary step.

6.4.2 Euclidean and Similarity Transformations

The optimal transformation problem for $\mathcal{G} = \mathcal{E}$, the group of Euclidean transformations, and $d = L_2^2$ is

$$\min_{(R,t)\in\mathcal{E}} \sum_{k=1}^{N} c_k ||a_k - (Rb_k + t)||_2^2,$$

where R is a rotation matrix. For a fixed R, the optimal translation must be $t^*(R) = \overline{a} - R\overline{b}$. Thus, the optimal Euclidean problem reduces to

$$\min_R \sum_{k=1}^{N} ||\widehat{a}_k - R\widehat{b}_k||_2^2 = \min_R ||\widehat{A} - R\widehat{B}||_{\mathrm{F}}^2, \tag{6.35}$$

where the columns of $\widehat{A}$ and $\widehat{B}$ are the vectors $\widehat{a}_k = \sqrt{c_k}(a_k - \overline{a})$ and $\widehat{b}_k = \sqrt{c_k}(b_k - \overline{b})$, and $||\cdot||_{\mathrm{F}}$ denotes the Frobenius matrix norm ([25]). Here we have also used the assumption that the c_k are nonnegative. The best rotation problem (6.35) is solved completely in [81]. The minimization problem (6.35) is easier to solve if we only require that R is orthogonal (i.e. we drop the requirement $\det(R) = 1$). Under this assumption, (6.35) is known as the *orthogonal Procrustes problem* ([25]). If $U\Sigma V^T$ is an SVD of $\widehat{A}\widehat{B}^T$, then the minimum value is $||\widehat{A}||_{\mathrm{F}}^2 + ||\widehat{B}||_{\mathrm{F}}^2 - 2\mathrm{tr}(\Sigma)$, and is achieved at $R = UV^T$.

The optimal transformation problem for $\mathcal{G} = \mathcal{S}$, the set of similarity transformations, allows for an additional scaling factor:

$$\min_{(s,R,t)\in\mathcal{S}} \sum_{k=1}^{N} c_k ||a_k - (sRb_k + t)||_2^2.$$

The special case of this problem in which $c_k \equiv 1/N$ is solved in [81] using the solution to (6.35). It is not difficult to repeat the analysis for general c_k and solve the optimal similarity problem as we have posed it, but we omit the details.

6.4.3 Linear and Affine Transformations

Finally, we consider the optimal transformation problem for linear and affine transformations with the L_2-distance squared. When $\mathcal{G} = \mathcal{L}$, the set of linear transformations, the optimal transformation problem becomes

$$\min_{L\in\mathcal{L}} \sum_{k=1}^{N} c_k ||a_k - Lb_k||_2^2.$$

Assuming that $c_k \geq 0$, an equivalent formulation is

$$\min_{L \in \mathcal{L}} \sum_{k=1}^{N} ||\widehat{a}_k - L\widehat{b}_k||_2^2 = \min_{L \in \mathcal{L}} ||\widehat{A} - L\widehat{B}||_{\mathrm{F}}^2,$$

where the columns of the matrices $\widehat{A}$ and $\widehat{B}$ are the vectors $\widehat{a}_k = \sqrt{c_k}a_k$ and $\widehat{b}_k = \sqrt{c_k}b_k$. An optimal linear transformation is $L^* = \widehat{A}\widehat{B}^\dagger$, where $\widehat{B}^\dagger$ is the pseudo-inverse ([25]) of $\widehat{B}$. In the case $\mathcal{G} = \mathcal{A}$, the set of affine transformations, the optimal transformation problem allows for an additional translation:

$$\min_{(L,t) \in \mathcal{A}} \sum_{k=1}^{N} c_k ||a_k - (Lb_k + t)||_2^2.$$

The optimal translation for a fixed L is $t^*(L) = \overline{a} - L\overline{b}$. Hence, with $\widetilde{a}_k = a_k - \overline{a}$ and $\widetilde{b}_k = b_k - \overline{b}$,

$$\min_{(L,t) \in \mathcal{A}} \sum_{k=1}^{N} c_k ||a_k - (Lb_k + t)||_2^2 = \min_{L \in \mathcal{L}} \sum_{k=1}^{N} c_k ||\widetilde{a}_k - L\widetilde{b}_k||_2^2,$$

and the affine problem reduces to the linear problem.

6.5 Allowing Weight-Altering Transformations

When a transformation g changes the weights of the distributions that it acts upon, in general $\mathcal{F}(\mathbf{x}, \mathbf{y}) \neq \mathcal{F}(\mathbf{x}, g(\mathbf{y}))$. This is because the constraints that define the feasible flows between two distributions depend on the weights in the distributions. Recall that there were two steps in proving that WORK sequence is decreasing when distribution weights are unchanged: (1) $F^{(k+1)}$ is a better flow for $g^{(k+1)}$ than the flow $F^{(k)}$, and (2) $g^{(k+1)}$ is a better transformation for $F^{(k)}$ than the transformation $g^{(k)}$. The inequality (6.14) which expresses step (2) still holds when distribution weights are not fixed. This is because $g^{(k+1)}$ is optimal for flow $F^{(k)}$ over all allowable transformations, and $g^{(k)}$ is one of the allowable transformations. The inequality (6.13) which expresses step (1), however, may not hold when distribution weights are changed. The flow $F^{(k+1)}$ is optimal for transformation $g^{(k+1)}$ over all flows in $\mathcal{F}(\mathbf{x}, g^{(k+1)}(\mathbf{y}))$, but flow $F^{(k)}$ may not be in the set $\mathcal{F}(\mathbf{x}, g^{(k+1)}(\mathbf{y}))$ – the flow $F^{(k)}$ was chosen from the set $\mathcal{F}(\mathbf{x}, g^{(k)}(\mathbf{y}))$.

It is easy to see that inequality (6.13) will hold if

$$\mathcal{F}(\mathbf{x}, g^{(k+1)}(\mathbf{y})) \supseteq \mathcal{F}(\mathbf{x}, g^{(k)}(\mathbf{y})), \tag{6.36}$$

for then $F^{(k)} \in \mathcal{F}(\mathbf{x}, g^{(k)}(\mathbf{y}))$ implies $F^{(k)} \in \mathcal{F}(\mathbf{x}, g^{(k+1)}(\mathbf{y}))$. Thus when we have an "increasing" sequence of feasible regions as specified by condition (6.36), we are guaranteed to get a decreasing WORK sequence. This, however, is not the end of the story because we are really after a decreasing EMD sequence. Remember that $\mathrm{EMD}^{(k)}$ is equal to $\mathrm{WORK}^{(k)}$ divided by the smaller of the total weights of $\mathbf{x}$ and $g^{(k)}(\mathbf{y})$, and the weight $g^{(k)}(\mathbf{y})$ is no longer constant over k.

The problem outlined above is that the WORK and the EMD sequences are not related by a constant multiplicative factor. We can get around this problem by a change of variables that moves the minimum total weight normalization factor into the definition of the flow. An example of such a change of variables has already been given in section 4.5 on scale estimation, where the change of variables is $\widehat{f}_{ij} = f_{ij}/c$ and c is the total weight of the lighter of the two distributions being compared (in section 4.5, we used h_{ij} instead of $\widehat{f}_{ij}$ as the new variables). This change of variables yielded a collection of transportation problems (one for each c) with increasing feasible regions $\mathcal{F}((X, w/c), \mathbf{y})$ as c is decreased. It followed that $E(c)$, the EMD between $\mathbf{x}$ and the transformed $\mathbf{y}$ as a function of the transformation parameter c, decreases as c decreases. In this case, the distribution transformations are transformations that scale down all the weights in the distribution by a factor c and leave the distribution points unchanged.

The same change of variables allows the FT iteration to be applied with some sets of transformations which alter both the weights and the points of distributions. Such transformations may be needed, for example, if a distribution point contains the position of an image region with some property and the corresponding weight is the region area; applying a similarity transformation with non-unit scale to region positions causes a change in region areas. Next we show how to apply the FT iteration in this case, where a distribution point is (L, a, b, x, y) in a combined CIE-Lab color space and image position space. The feature point (L, a, b, x, y) with weight w is meant to indicate that there is a region in the image plane with area w that has centroid (x, y) and color (L, a, b).

We denote the distribution summaries of the image and the pattern as

$$\begin{aligned} \mathbf{x} &= \{\, (x_1, w_1), \ldots, (x_m, w_m) \,\} = \{\, ((a_1, p_1), w_1), \ldots, ((a_m, p_m), w_m) \,\}, \quad \text{and} \\ \mathbf{y} &= \{\, (y_1, u_1), \ldots, (y_n, u_n) \,\} = \{\, ((b_1, q_1), u_1), \ldots, ((b_n, q_n), u_n) \,\}, \end{aligned}$$

respectively, where $x_i = (a_i, p_i)$ divides feature point x_i into its color components a_i and position components p_i, and $y_j = (b_j, q_j)$ divides feature point y_j into its color components b_j and position components q_j. We assume that the ground distance d_{cp} in the combined color-position space is given by

$$d_{\mathrm{cp}}(x,y) = d_{\mathrm{cp}}((a,p),(b,q)) = d_{L_2}(a,b) + \lambda d_{L_2^2}(p,q),$$

where the parameter λ trades off between distance in color space and distance in position space.[7] We also assume that the weight normalization $w_\Sigma = u_\Sigma = 1$. Finally, we denote similarity transformations by $g = (s, \theta, t)$. The action of g on distribution $\mathbf{y}$ is defined by

$$g((b_j, q_j), u_j) = ((b_j, sR_\theta q_j + t), \kappa s^2 u_j),$$

where κ is a constant factor relating the scale s in positional units to the corresponding scale in area units (which need not be exactly the square of the position units since we have assumed $u_\Sigma = 1$). In the analysis that follows, we define the area scale $c = \kappa s^2$.

Our EMD under similarity transformation problem is

$$\mathrm{EMD}_{\mathcal{S}}(\mathbf{x},\mathbf{y}) = \min_{g\in\mathcal{S}} \min_{F\in\mathcal{F}(\mathbf{x},g(\mathbf{y}))} \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij}(d_{L_2}(a_i,b_j) + \lambda d_{L_2^2}(p_i, g(q_j)))}{c}.$$

If we let $\widehat{f}_{ij} = f_{ij}/c$, then the problem becomes

$$\mathrm{EMD}_{\mathcal{S}}(\mathbf{x},\mathbf{y}) = \min_{g\in\mathcal{S}} \min_{\widehat{F}\in\widehat{\mathcal{F}}(\mathbf{x},g(\mathbf{y}))} \sum_{i=1}^m \sum_{j=1}^n \widehat{f}_{ij}(d_{L_2}(a_i,b_j) + \lambda d_{L_2^2}(p_i, g(q_j))),$$

where the feasible region $\widehat{\mathcal{F}}(\mathbf{x}, g(\mathbf{y})) = \mathcal{F}((X, w/c), \mathbf{y})$ as defined in section 4.5 by conditions (4.17), (4.18), and (4.19).

The minimization problems to be solved by the FT iteration are

$$\widehat{F}^{(k)} = \arg\left(\min_{\widehat{F}\in\widehat{\mathcal{F}}(\mathbf{x},g^{(k)}(\mathbf{y}))} \sum_{i=1}^m \sum_{j=1}^n \widehat{f}_{ij}(d_{L_2}(a_i,b_j) + \lambda d_{L_2^2}(p_i, g^{(k)}(q_j)))\right), \quad (6.37)$$

$$g^{(k+1)} = \arg\left(\min_{g\in\mathcal{S}} \sum_{i=1}^m \sum_{j=1}^n \widehat{f}_{ij}^{(k)}(d_{L_2}(a_i,b_j) + \lambda d_{L_2^2}(p_i, g(q_j)))\right). \quad (6.38)$$

Since g does not change the color component of a distribution point, we can compute $g^{(k+1)}$

[7] If colors are represented in CIE-Lab space, then the Euclidean distance is the natural choice for a distance in color space. The analysis that follows, however, does not require the distance in color space to be L_2.

defined in equation (6.38) by solving

$$g^{(k+1)} = \arg\left(\min_{g\in\mathcal{S}} \sum_{i=1}^{m}\sum_{j=1}^{n} \widehat{f}_{ij}^{(k)} d_{L_2^2}(p_i, g(q_j))\right). \tag{6.39}$$

Section 6.4.2 discusses the solution to this optimal similarity transformation problem. Solving for $\widehat{F}^{(k)}$ in (6.37) is still a transportation problem.

If we define

$$\mathrm{EMD}^{(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} \widehat{f}_{ij}^{(k)} (d_{L_2}(a_i, b_j) + \lambda d_{L_2^2}(p_i, g^{(k)}(q_j))),$$

then, as previously argued, we will get a decreasing EMD sequence if we can guarantee

$$\widehat{\mathcal{F}}(\mathbf{x}, g^{(k+1)}(\mathbf{y})) \supseteq \widehat{\mathcal{F}}(\mathbf{x}, g^{(k)}(\mathbf{y})) \quad \forall k. \tag{6.40}$$

If $g^{(k)} = (s^{(k)}, \theta^{(k)}, t^{(k)})$, then (6.40) will hold if $s^{(k+1)} \leq s^{(k)}$ $\forall k$. So the FT iteration will yield a decreasing EMD sequence if we only allow scale to decrease as the iteration proceeds. For the specific case of allowing a similarity transformation, this can be accomplished as follows. In computing $g^{(k+1)}$, first perform the minimization over the set of similarity transformations as in (6.39) to get a similarity transformation $(s^{(k+1)}, \theta^{(k+1)}, t^{(k+1)})$. If $s^{(k+1)} \leq s^{(k)}$, then set $g^{(k+1)} = (s^{(k+1)}, \theta^{(k+1)}, t^{(k+1)})$. Otherwise, set $g^{(k+1)}$ to the best Euclidean transformation with fixed scale $s^{(k)} : g^{(k+1)} = (s^{(k)}, \theta^{(k+1)}, t^{(k+1)})$ where

$$(\theta^{(k+1)}, t^{(k+1)}) = \arg\left(\min_{g\in\mathcal{E}} \sum_{i=1}^{m}\sum_{j=1}^{n} \widehat{f}_{ij}^{(k)} d_{L_2^2}(p_i, g(s^{(k)} q_j))\right).$$

This optimal Euclidean transformation problem is discussed in section 6.4.2.

6.6 Some Specific Cases

There are some specific cases of transformation set, ground distance function, and feature space that are worth mentioning in our discussion of the EMD under transformation sets.

6.6.1 The Equal-Weight EMD under Translation with $d = L_2^2$

Recall from section 6.4.1.1 that if d is the L_2-distance squared, then the unique optimal translation for a fixed flow is given by the centroid difference

$$t^*_{L_2^2} = \overline{a} - \overline{b} = \frac{\sum_{k=1}^N c_k a_k}{c_\Sigma} - \frac{\sum_{k=1}^N c_k b_k}{c_\Sigma},$$

where the a_k, b_k, and c_k are defined by the distribution points and the fixed flow as in (6.17), (6.18), and (6.19). In terms of the original points and flow vector,

$$t^*_{L_2^2} = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} x_i}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} - \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} y_j}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}}. \tag{6.41}$$

If $\mathbf{x}$ and $\mathbf{y}$ are equal-weight distributions (and $\gamma = 1$), then $\sum_{i=1}^m f_{ij} = u_j$, $\sum_{j=1}^n f_{ij} = w_i$, and $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = w_\Sigma = u_\Sigma$ for any feasible flow $F = (f_{ij})$. Using these facts in equation (6.41) shows that the best translation for any feasible flow $F = (f_{ij})$ is $t^*_{L_2^2} = \overline{\mathbf{x}} - \overline{\mathbf{y}}$. Therefore, the FT iteration described in section 6.3 is not needed in the equal-weight case to compute $\text{EMD}_{\mathcal{T},L_2^2}(\mathbf{x},\mathbf{y})$. Instead, simply translate $\mathbf{y}$ by $\overline{\mathbf{x}} - \overline{\mathbf{y}}$ (this lines up the centroids of $\mathbf{x}$ and $\mathbf{y}$) to get $\widehat{\mathbf{y}}$ and compute $\text{EMD}(\mathbf{x}, \widehat{\mathbf{y}})$.

6.6.2 The Equal-Weight EMD under Translation on the Real Line

In this section, we assume that the ground distance is the absolute value between points on the real line ($d = L_1$). Recall the definition in section 4.3.2 of the CDF flow F^{CDF} between two equal-weight distributions $\mathbf{x} = (X, w)$ and $\mathbf{y} = (Y, u)$ on the real line:

$$f_{ij}^{\text{CDF}} = |[W_{i-1}, W_i] \cap [U_{j-1}, U_j]|,$$

where

$$\begin{array}{rcrcll} W_k & = & W(x_k) & = & \sum_{i=1}^k w_i & \text{and} \\ U_l & = & U(y_l) & = & \sum_{j=1}^l u_j. & \end{array}$$

Here the points and corresponding weights in the distributions are numbered according to increasing position along the real line: $x_1 < \cdots < x_m$ and $y_1 < \cdots < y_n$. In Theorem 5, we showed that the CDF flow F^{CDF} is an optimal flow between $\mathbf{x}$ and $\mathbf{y}$ if $d = L_1$. Now denote the translation of $\mathbf{y}$ by t as

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

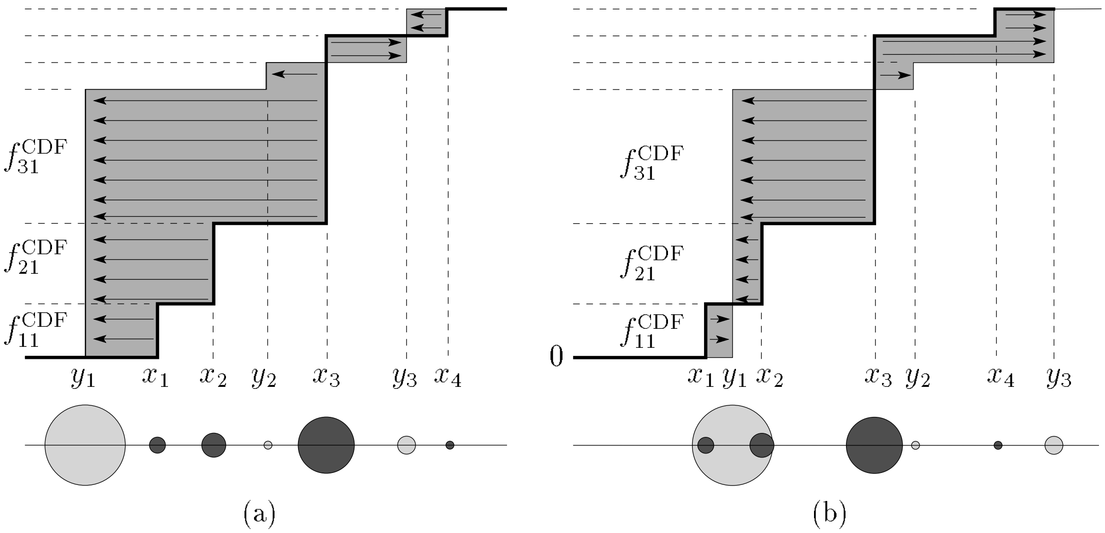

Figure 6.4: The Equal-Weight EMD under Translation in 1D with $d = L_1$. The same flow F^{CDF} is optimal for (a) $\mathbf{x}$ and $\mathbf{y}$, and (b) $\mathbf{x}$ and $\mathbf{y} \oplus t$. We re-use the labels y_j in (b) instead of using $y_j + t$ in order to make all the labels fit in the given space.

Since the sorted order of the points of $\mathbf{y} \oplus t$ is the same as the sorted order of the points of $\mathbf{y}$, and the weights of $\mathbf{y} \oplus t$ are the same as the weights of $\mathbf{y}$, the CDF flow between $\mathbf{x}$ and $\mathbf{y}$ is the same as the CDF flow between $\mathbf{x}$ and $\mathbf{y} \oplus t$. By Theorem 5, this CDF flow is also an optimal flow between $\mathbf{x}$ and $\mathbf{y} \oplus t$. See Figure 6.4 for an example.

Now for fixed t, the optimal transformation step (6.11) in the FT iteration is to compute

$$\delta(t) = \min_{F \in \mathcal{F}(\mathbf{x},\mathbf{y})} \mathrm{WORK}(F^{\mathrm{CDF}}, \mathbf{x}, \mathbf{y} \oplus t).$$

Since the CDF flow is optimal for every $t \in \mathbf{R}$,

$$\delta(t) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}^{\mathrm{CDF}} |x_i - (y_j + t)|.$$

Rewriting the 2D index as a 1D index, we have

$$\delta(t) = \sum_{k=1}^{N} f_k^{\mathrm{CDF}} |z_k - t|.$$

Functions of this form were studied extensively in section 6.4.1.2 where we gave the solution to this “minisum problem” on the line. This solution gives us the EMD under translation since

$$\mathrm{EMD}_{\mathcal{T}}(\mathbf{x},\mathbf{y}) = \frac{\min_{F\in\mathcal{F}(\mathbf{x},\mathbf{y}),t\in\mathbf{R}} \mathrm{WORK}(F^{\mathrm{CDF}},\mathbf{x},\mathbf{y}\oplus t)}{\min(w_\Sigma,u_\Sigma)} = \frac{\min_{t\in\mathbf{R}}\delta(t)}{\min(w_\Sigma,u_\Sigma)}.$$

The function $\delta(t)$ is piecewise linear with monotonic slope increasing from a negative value at $t=-\infty$ to a positive value at $t=+\infty$ (see Figures 6.2 and 6.3). Thus, $\delta(t)$ is convex and has its minimum at a point t at which the slope first becomes nonnegative.

Once the m points in $\mathbf{x}$ and the n points in $\mathbf{y}$ have been sorted, the CDF flow F^{CDF} can be computed in $\Theta(m+n)$ time using the second algorithm labelled EMD_1 given on page 78 in section 4.3.2. The result of this algorithm is an array of $\Theta(m+n)$ records containing a pair (i,j) and the flow value f_{ij}. Any pair (i,j) not appearing in this array has flow value $f_{ij}=0$. To compute the optimal translation (and the actual value of the EMD under translation) using the results in section 6.4.1.2, we need to find the first index k at which the slope m_k,

$$\begin{aligned} m_0 &= -\sum_{k=1}^{N} f_k^{\mathrm{CDF}} = -w_\Sigma = -u_\Sigma, \\ m_{k+1} &= m_k + 2f_{k+1}^{\mathrm{CDF}}, \end{aligned}$$

is greater than or equal to zero. This can be done by sorting the returned flow pairs (i,j) by increasing 1D index k in time $\Theta((m+n)\log(m+n))$, and then tracking the slope m_k while marching through the previously sorted flow array in time $O(m+n)$ (the array traversal can stop once it reaches k such that $m_k\geq 0$). This algorithm to compute the EMD under translation between equal-weight distributions on the line requires $\Theta((m+n)\log(m+n))$ time.

6.6.3 The Equal-Weight EMD under $\mathcal{G}$ with $m=n=2$

In this section, we consider the problem of matching equal-weight distributions $\mathbf{x}$ and $\mathbf{y}$ with two points each ($m=n=2$) under a general transformation set $\mathcal{G}$, where $g\in\mathcal{G}$ changes only the points in a distribution. Without loss of generality, we assume the unit-weight normalizations $w_\Sigma=u_\Sigma=1$.

The conditions which define the feasible flow set $\mathcal{F}(\mathbf{x},\mathbf{y})=\mathcal{F}(\mathbf{x},g(\mathbf{y}))$ are $f_{11}\geq 0$, $f_{12}\geq 0$, $f_{21}\geq 0$, $f_{22}\geq 0$, and

$$f_{11}+f_{12} = w_1, \tag{6.42}$$

$$f_{21}+f_{22} = w_2 = 1-w_1, \tag{6.43}$$

$$
\begin{aligned}
f_{11} + f_{21} &= u_1, &(6.44)\\
f_{12} + f_{22} &= u_2 = 1 - u_1. &(6.45)
\end{aligned}
$$

Using (6.42)–(6.45), we can write all flow variables in terms of f_{11}:

$$
\left.\begin{aligned}
f_{12} &= w_1 - f_{11}\\
f_{21} &= u_1 - f_{11}, \quad \text{and}\\
f_{22} &= 1 - w_1 - f_{21} = 1 - (w_1 + u_1) + f_{11}.
\end{aligned}\right\} \qquad (6.46)
$$

From (6.46), we see that

$$
\max(0, (w_1 + u_1) - 1) \leq f_{11} \leq \min(u_1, w_1) \qquad (6.47)
$$

is a necessary condition for $F = (f_{ij})$ to be feasible since flow variables must be nonnegative. Also, every f_{11} which satisfies (6.47) defines a feasible flow F according to equations (6.46). Thus, we have argued that the set of feasible flows $\mathcal{F}(\mathbf{x}, \mathbf{y})$ is also defined by conditions (6.46) and (6.47).[8]

Using (6.46), we may write the work done by F to match $\mathbf{x}$ and $\mathbf{y}$ as

$$
\begin{aligned}
\text{WORK}(F, \mathbf{x}, \mathbf{y}) &= f_{11}d_{11} + f_{12}d_{12} + f_{21}d_{21} + f_{22}d_{22}\\
&= ((d_{11} + d_{22}) - (d_{12} + d_{21}))f_{11}\\
&\quad + (w_1 d_{12} + u_1 d_{21} + (1 - (w_1 + u_1))d_{22}).
\end{aligned}
$$

When we allow a transformation $g \in \mathcal{G}$, the point distances d_{ij} become functions $d_{ij}(g)$ of g, and we have

$$
\begin{aligned}
\text{WORK}(F, \mathbf{x}, g(\mathbf{y})) &= ((d_{11}(g) + d_{22}(g)) - (d_{12}(g) + d_{21}(g)))f_{11} &(6.48)\\
&\quad + (w_1 d_{12}(g) + u_1 d_{21}(g) + (1 - (w_1 + u_1))d_{22}(g)).
\end{aligned}
$$

Since $w_\Sigma = u_\Sigma = 1$,

$$
\text{EMD}_{\mathcal{G}}(\mathbf{x}, \mathbf{y}) = \min_{F \in \mathcal{F}(\mathbf{x},\mathbf{y}), g \in \mathcal{G}} \text{WORK}(F, \mathbf{x}, g(\mathbf{y})). \qquad (6.49)
$$

From (6.48), we see that the minimum in (6.49) must be achieved at one of two feasible

[8] Note that $\mathcal{F}(\mathbf{x}, \mathbf{y})$ defined by (6.47) is nonempty. Since distribution weights are nonnegative, we have $\min(u_1, w_1) \geq 0$. Without loss of generality, suppose $w_1 \leq u_1$. Then $\min(u_1, w_1) - ((w_1 + u_1) - 1) = 1 - u_1 \geq 0$ (since $u_\Sigma = 1$, $u_1 \geq 0$ implies $u_1 \leq 1$), and hence $\min(u_1, w_1) \geq \max(0, (w_1 + u_1) - 1)$. The case $u_1 \leq w_1$ is similar.

flows F^1 or F^2. More precisely, an optimal F for a given g is

$$f_{11} = \begin{cases} f_{11}^1 = \min(u_1, w_1) & \text{if } d_{11}(g) + d_{22}(g) < d_{12}(g) + d_{21}(g) \\ f_{11}^2 = \max(0, (u_1 + w_1) - 1) & \text{if } d_{11}(g) + d_{22}(g) \geq d_{12}(g) + d_{21}(g) \end{cases},$$

where f_{12}, f_{21}, and f_{22} follow from (6.46).

Since we know that the minimum in (6.49) is achieved at one of the flows F^1 or F^2 given above, we can compute

$$\mathrm{EMD}_{\mathcal{G}}(\mathbf{x}, \mathbf{y}) = \min\left(\min_{g \in \mathcal{G}} \mathrm{WORK}(F^1, \mathbf{x}, g(\mathbf{y})), \min_{g \in \mathcal{G}} \mathrm{WORK}(F^2, \mathbf{x}, g(\mathbf{y})) \right)$$

by solving an optimal transformation problem for each of F^1 and F^2.

6.7 Global Convergence in $\mathcal{F} \times \mathcal{G}$?

This section is devoted to the following question: Under what conditions does the FT iteration converge to the global minimum of $\mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y})) : \mathcal{F}(\mathbf{x}, \mathbf{y}) \times \mathcal{G} \longrightarrow \mathbf{R}_{\geq 0}$? There are many parameters here, including (1) the transformation set $\mathcal{G}$, (2) the ground distance d, (3) the dimension of the underlying points, (4) whether or not $\mathbf{x}$ and $\mathbf{y}$ are equal-weight distributions, and (5) the distributions $\mathbf{x}$ and $\mathbf{y}$ themselves. Here we shall only consider transformations which modify the distribution locations but not their corresponding weights.

In section 6.7.1, we consider the problem for unequal-weight distributions $\mathbf{x}$ and $\mathbf{y}$. We call this the "partial matching" case because some of the weight in the heavier distribution will be unmatched. In different regions of $\mathcal{G}$, different parts of the heavier distribution may be used in an optimal flow, and this makes it impossible to prove a guarantee of global convergence.

In section 6.7.2, we argue that the FT iteration is guaranteed to converge to the global minimum of $\mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y}))$ if either (1) there is a transformation g^* which is the unique optimal transformation for every feasible flow, or (2) there is a feasible flow F^* which is the unique optimal flow for every transformation. These may seem like highly constrained situations, but we have already encountered an example of (1), namely the EMD under translation between equal-weight distributions with ground distance $d = L_2^2$. We also discuss the effect of removing the uniqueness requirement from (1) and (2).

In section 6.7.3, we consider the case of matching a distribution to a translated version of itself. The EMD under translation is obviously zero in this perfect matching case. We briefly describe experiments which show that in practice the FT iteration converges to the global minimum of zero.

In section 6.7.4, we demonstrate that there can be transformations which are locally but not globally optimal even for equal-weight comparisons. We give an example of equal-weight distributions in the plane with two points each ($m = n = 2$) for which there are local minima in $\mathcal{F} \times \mathcal{T}$ for the L_2 and L_1 ground distances. We also show that the WORK function with the L_2^2 distance can have local minima if $\mathcal{G}$ is the group of rotations.

If there are local minima in $\mathcal{F} \times \mathcal{G}$, then it is hard to have a guarantee of convergence to the global minimum. If there is a local minimum at (F^0, g^0), then g^0 is locally optimal for F^0, and F^0 is locally optimal for g^0. But holding $g = g^0$ fixed yields a WORK function which is linear in F, so a locally optimal flow for g^0 must be a globally optimal flow for g^0. In many cases there are no local minima of the WORK function in g when F is held fixed (e.g. when $\mathcal{G} = \mathcal{T}$, $d = L_p$), so a locally optimal transformation for F^0 is a globally optimal transformation for F^0. We have already seen that the FT iteration gets stuck at (F, g) when F and g are mutually optimal for each other.

6.7.1 Partial Matching

When one distribution is heavier than the other, some of the mass in the heavier distribution is unmatched in a feasible flow. In different regions of $\mathcal{G}$, different parts of the heavier distribution may be used in an optimal flow. This fact allows one to develop examples that possess local minima.

Imagine a distribution $\mathbf{x}$ composed of two spatially separated sub-distributions $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$ which are equal-weight, say $\frac{1}{2}w_\Sigma$ each. Now consider matching $\mathbf{x}$ to a distribution $\mathbf{y}$ with weight $u_\Sigma = \frac{1}{2}w_\Sigma$. Of all the transformations g which place $g(\mathbf{y})$ in the $\widehat{\mathbf{x}}$ part of the point space, there will be an optimal transformation $\widehat{g}^*$. If the sub-distributions $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$ are separated enough, the corresponding flows will not involve any mass from $\widetilde{\mathbf{x}}$. Similarly, of all the transformations g which place $g(\mathbf{y})$ in the $\widetilde{\mathbf{x}}$ part of the point space, there will be an optimal transformation $\widetilde{g}^*$. Assuming that $\widehat{g}^*(\mathbf{y})$ does not match $\widehat{\mathbf{x}}$ equally as well as $\widetilde{g}^*(\mathbf{y})$ matches $\widetilde{\mathbf{x}}$, one of $\widehat{g}^*$ and $\widetilde{g}^*$ is only a locally optimal transformation. Figures 6.5 and 6.6 show examples in 1D and 2D, respectively.

We can also create examples in which there are as many local minima as we like if we allow the ratio u_Σ / w_Σ to be arbitrarily small. If $\mathbf{x}$ is L well-separated copies $\mathbf{y} \oplus t_l$ of $\mathbf{y}$, then $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t_l) = 0$ for $l = 1, \ldots, L$. We can produce $\geq L - 1$ only locally optimal translations by slightly perturbing the points in each copy of $\mathbf{y}$. In general, there may be no overlap in the mass of the heavier distribution used to match the mass in the lighter distribution in different parts of the transformation space.

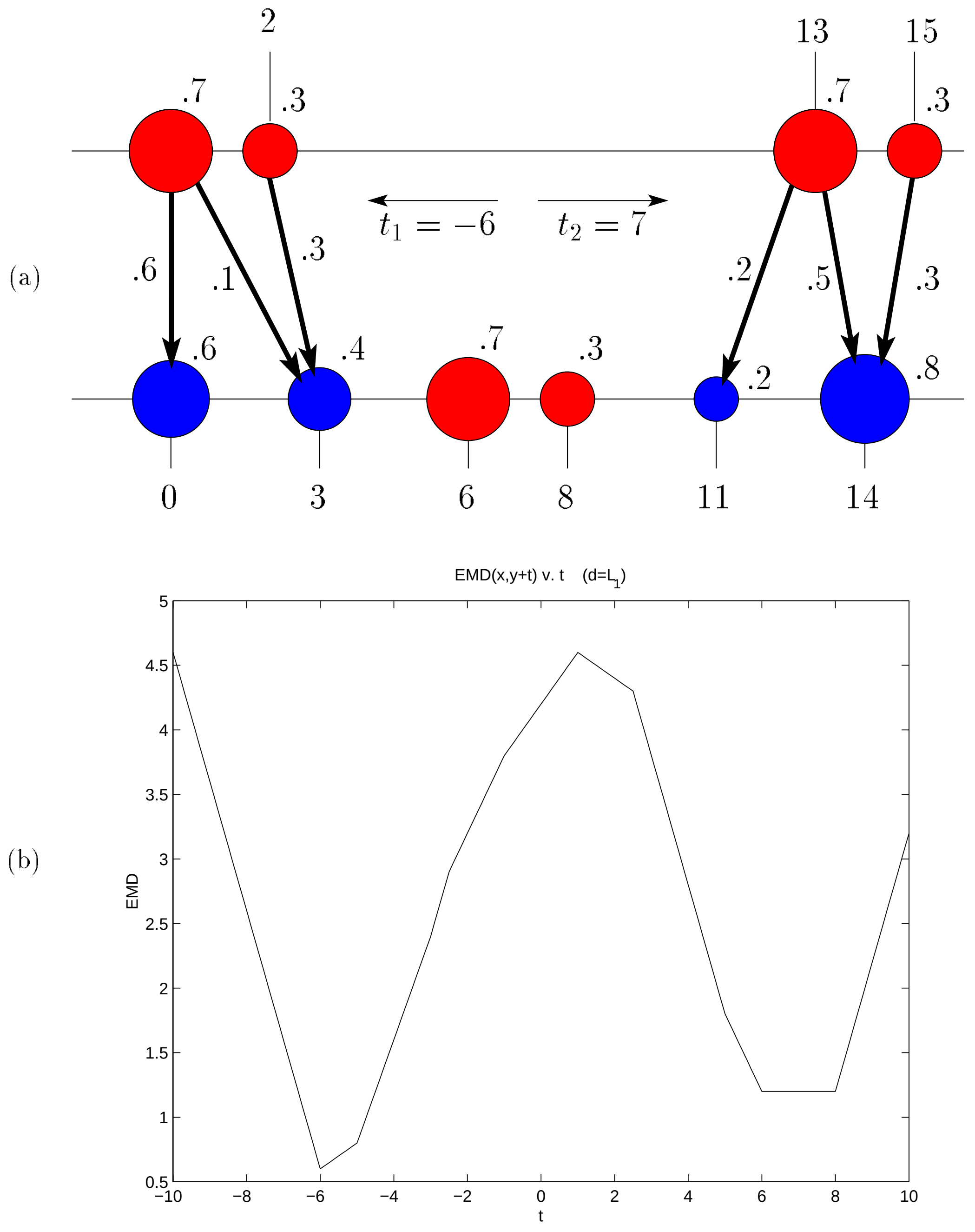

Figure 6.5: A Local Minimum in a 1D Partial Matching Case. (a) Distributions over the real line $\mathbf{x}$ and $\mathbf{y}$ are shown on the bottom line in blue and red, respectively. Translating $\mathbf{y}$ by $t_1 = -6$ gives $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t_1) = .6(0) + .1(3) + .3(1) = .6$, while translating $\mathbf{y}$ by $t_2 = 7$ gives $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t_2) = .2(2) + .5(1) + .3(1) = 1.2$. The translation t_1 yields the global minimum, while the translation t_2 yields a local minimum as one can see from (b) the graph of $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t)$ v. t.

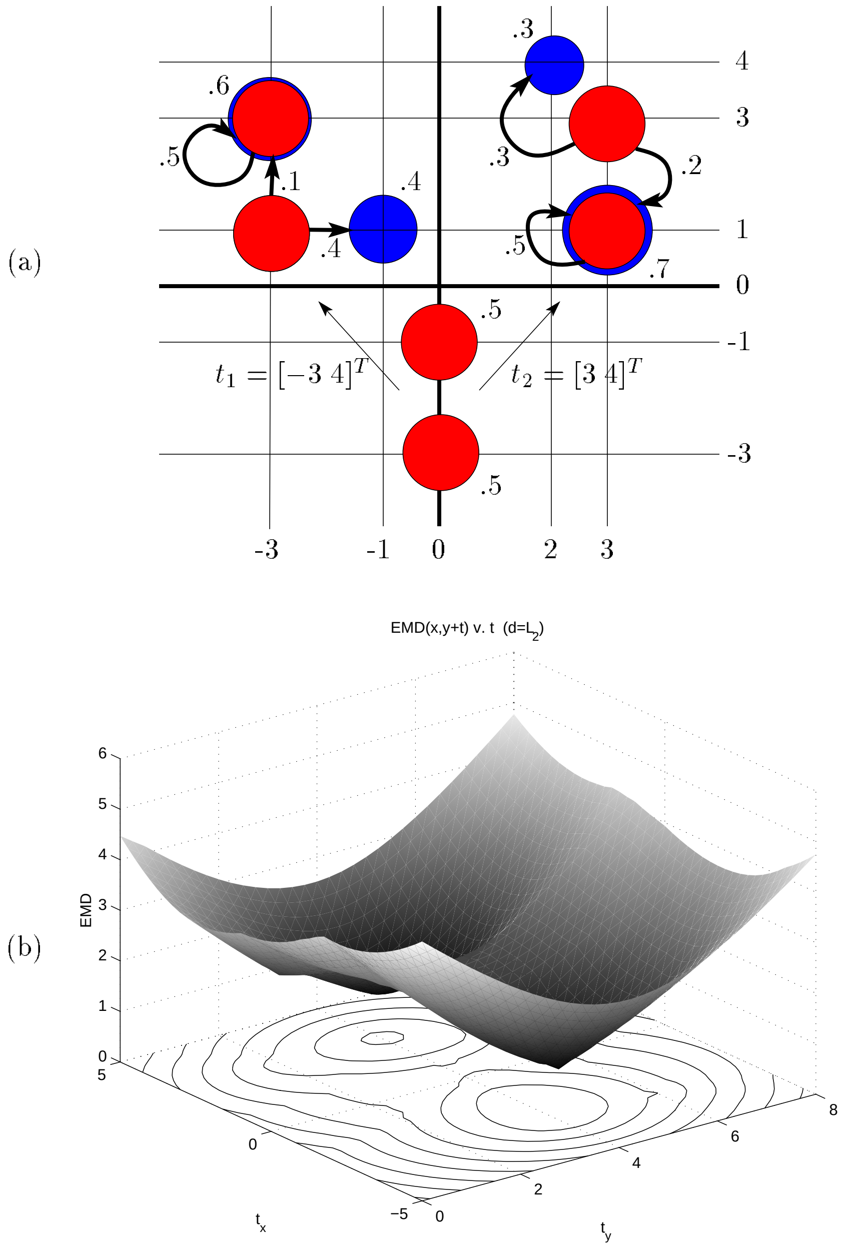

Figure 6.6: A Local Minimum in a 2D Partial Matching Case. (a) Distributions over the plane $\mathbf{x}$ and $\mathbf{y}$ are shown in blue and red, respectively. Translating $\mathbf{y}$ by $t_1 = [-3\ 4]^T$ gives $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t_1) = .5(0) + .1(2) + .4(2) = 1.0$, while translating $\mathbf{y}$ by $t_2 = [3\ 4]^T$ gives $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t_2) = .5(0) + .3(\sqrt{2}) + .2(2) \doteq .824$. The translation t_2 yields the global minimum, while the translation t_1 yields a local minimum as one can see from (b) the graph of $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t)$ v. t.

6.7.2 One Optimal Flow or Transformation

If g^* is the unique optimal transformation for every feasible flow, then the WORK sequence converges to the global minimum work value in only a couple of iterations:

$$g^{(0)} \longrightarrow F^{(0)} \longrightarrow g^{(1)} = g^* \longrightarrow F^{(1)} \longrightarrow g^{(2)} = g^* \longrightarrow F^{(2)},$$

where $\text{WORK}(F^{(1)}, \mathbf{x}, g^{(1)}(\mathbf{y})) = \text{WORK}(F^{(1)}, \mathbf{x}, g^{(2)}(\mathbf{y})) = \text{WORK}(F^{(2)}, \mathbf{x}, g^{(2)}(\mathbf{y}))$ is the global minimum. We have already encountered such a case. For equal-weight distributions with $\mathcal{G} = \mathcal{T}$ and $d = L_2^2$, $t^* = \bar{x} - \bar{y}$ is the unique optimal translation for every feasible flow.

If there is a unique optimal flow F^* for every transformation, then the WORK sequence also converges to the global minimum work value in only a couple of iterations:

$$g^{(0)} \longrightarrow F^{(0)} = F^* \longrightarrow g^{(1)} \longrightarrow F^{(1)} = F^* \longrightarrow g^{(2)} \longrightarrow F^{(2)} = F^*,$$

where $\text{WORK}(F^{(1)}, \mathbf{x}, g^{(1)}(\mathbf{y})) = \text{WORK}(F^{(1)}, \mathbf{x}, g^{(2)}(\mathbf{y})) = \text{WORK}(F^{(2)}, \mathbf{x}, g^{(2)}(\mathbf{y}))$ is the global minimum. We have seen a case which comes close to meeting this requirement. For equal-weight distributions on the real line with $\mathcal{G} = \mathcal{T}$ and $d = L_1$, the CDF flow F^{CDF} is optimal for every $t \in \mathcal{T}$, although it is not necessarily the unique optimal flow for every $t \in \mathcal{T}$. We have been unable to rule out (even in this specific case) the possibility that

- $\widehat{F}$ and F^* are both optimal for some $g^{(k)} = \widehat{g}$,
- $\widehat{F}$ is returned by the transportation problem solver instead of F^*,
- $\widehat{g}$ is optimal for $F^{(k)} = \widehat{F}$,
- $\widehat{g}$ is returned by an optimal transformation solver as optimal for $F^{(k)}$, and
- $(\widehat{F}, \widehat{g})$ is not a globally optimal (flow,transformation) pair.

In this case, the FT iteration converges to $(\widehat{F}, \widehat{g})$ which is not globally optimal. We have also been unable to rule out the analogous possibility in the case when one transformation g^* is optimal for for every flow, but g^* is not the unique optimal transformation for every flow.

Now suppose that F^* is an optimal flow for every transformation. If the FT iteration ever reaches a transformation $g^{(k)}$ for which F^* is the unique optimal flow, then the iteration will converge to the global minimum work value. Here we are guaranteed that $F^{(k)} = F^*$, and we argued in section 6.3.2 that the FT iteration converges to the global minimum if the flow sequence ever reaches a globally optimal flow. Similarly, if g^* is an optimal transformation for every flow, then the FT iteration converges to the global minimum if it ever reaches a flow $F^{(k)}$ for which g^* is the unique optimal transformation.

6.7.3 A Perfect Match under Translation

It is clear that $\mathrm{EMD}_{\mathcal{T}}(\mathbf{y}, \mathbf{y} \oplus \Delta y) = 0$, where the translation that best aligns $\mathbf{y}$ and $\mathbf{y} \oplus \Delta y$ is $t^* = -\Delta y$. Is the FT iteration guaranteed to converge to the global minimum in this perfect matching case? We begin exploring this question by considering the EMD between $\mathbf{y}$ and $\mathbf{y} \oplus \Delta y$ without allowing a translation.

Theorem 13 *The EMD between a distribution and a translation of the distribution is*

$$\mathrm{EMD}(\mathbf{y}, \mathbf{y} \oplus \Delta y) = \begin{cases} ||\Delta y||_p & \textit{if } d = L_p \ (p \geq 1)\textit{, and} \\ ||\Delta y||_2^2 & \textit{if } d = L_2^2 \end{cases} .$$

Proof. The flow F^{ID} defined by $f_{ij}^{\mathrm{ID}} = \delta_{ij} u_j$, where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$, gives normalized WORK values

$$\frac{\mathrm{WORK}(F^{\mathrm{ID}}, \mathbf{y}, \mathbf{y} \oplus \Delta y)}{u_\Sigma} = \begin{cases} ||\Delta y||_p & \text{if } d = L_p \ (p \geq 1)\text{, and} \\ ||\Delta y||_2^2 & \text{if } d = L_2^2 \end{cases} .$$

Since the EMD is the normalized WORK value for the optimal flow, we have

$$\mathrm{EMD}(\mathbf{y}, \mathbf{y} \oplus \Delta y) \leq \begin{cases} ||\Delta y||_p & \text{if } d = L_p \ (p \geq 1)\text{, and} \\ ||\Delta y||_2^2 & \text{if } d = L_2^2 \end{cases} . \tag{6.50}$$

Is there a feasible flow which requires less work than the size of Δy for either ground distance? The answer is "no". By the centroid lower bound theorems 6 and 7 (and the fact that $\overline{\mathbf{y} \oplus \Delta y} = \overline{\mathbf{y}} + \Delta y$), we know that

$$\mathrm{EMD}(\mathbf{y}, \mathbf{y} \oplus \Delta y) \geq \begin{cases} ||\Delta y||_p & \text{if } d = L_p \ (p \geq 1)\text{, and} \\ ||\Delta y||_2^2 & \text{if } d = L_2^2 \end{cases} . \tag{6.51}$$

The result follows from the opposite inequalities (6.50) and (6.51). ■

It is somewhat surprising that no matter how small or large the shift Δy, there is no better flow than matching a point to its translate.

If two points y_k and y_l have equal weights $u_k = u_l \equiv \alpha$, then F^{ID} will *not* be the unique optimal flow between $\mathbf{y}$ and $\mathbf{y} \oplus \Delta y$ if $\Delta y = y_l - y_k$ and $d = L_p$. In this case, the following slight modification $\widehat{F}$ of F^{ID} is also an optimal feasible flow:

$$\widehat{f}_{ij} = \begin{cases} u_i & \text{if } i = j,\ i \neq k,\ i \neq l, \\ u_k = u_l \equiv \alpha & \text{if } (i,j) = (k,l) \text{ or } (i,j) = (l,k), \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

When $d = L_p$ and $\Delta y = y_l - y_k$, we have

$$\begin{array}{lclcll} d_{kk} & = & d(y_k, y_k + \Delta y) & = & \|\Delta y\|_p, & \\ d_{ll} & = & d(y_l, y_l + \Delta y) & = & \|\Delta y\|_p, & \\ d_{kl} & = & d(y_k, y_l + \Delta y) & = & 2\|\Delta y\|_p, & \text{and} \\ d_{lk} & = & d(y_l, y_k + \Delta y) & = & 0. & \end{array}$$

In order to match the mass at y_k and y_l, the flow F^{ID} spends an amount of work equal to $\alpha d_{kk} + \alpha d_{ll} = 2\|\Delta y\|_p \alpha$, while the flow $\widehat{F}$ spends the same amount of work $\alpha d_{kl} + \alpha d_{lk} = 2\|\Delta y\|_p \alpha$ in a different way.

If we replace Δy by $\Delta y + t$, then we see that F^{ID} is an optimal flow between $\mathbf{y}$ and $(\mathbf{y} \oplus \Delta y) \oplus t = \mathbf{y} \oplus (\Delta y + t)$ for every translation t. Global convergence of the FT iteration is guaranteed when F^{ID} is the unique optimal flow for every t (see section 6.7.2). From the above discussion, however, we know that F^{ID} will *not* be the unique optimal flow for $t = (y_l - y_k) - \Delta y$ if $u_k = u_l$. On the other hand, F^{ID} is optimal for every t, so it is potentially returned as $F^{(k)}$ for every translation iterate $t^{(k)}$. F^{ID} might be returned even when it is not the unique optimal flow for $t^{(k)}$. Of course, once the FT iteration reaches a flow iterate $F^{(k)} = F^{\mathrm{ID}}$, the corresponding WORK sequence immediately converges to the global minimum WORK of zero.

The transportation simplex algorithm used to compute $F^{(k)}$ is an iterative algorithm. There are a few common rules for computing an initial feasible solution to the transportation problem, including the northwest corner rule, Vogel's method, and Russell's method ([32]). Applying the northwest corner rule to the transportation problem specified by $\mathbf{y}$ and $(\mathbf{y} \oplus \Delta y) \oplus t$ results in F^{ID} as the initial feasible solution for every t. Since F^{ID} is optimal, the transportation simplex algorithm will return F^{ID} for every t when the northwest corner rule is used. In this case, the FT iteration is guaranteed to converge to the global minimum.

The northwest corner rule is faster than Vogel's and Russell's methods, but it produces an initial solution which is usually not as close to optimal. Consequently, more iterations are usually required with the northwest corner rule. In general, the transportation simplex algorithm will find an optimal solution faster using Vogel's or Russell's method because fewer iterations will be required. In constrast to the northwest corner rule, these methods use the transportation problem costs (which are the ground distances in the EMD context) to compute an initial solution.

The EMD code that we use in our work applies Russell's method. The initial solution computed by this method is not necessarily F^{ID}, so there is no guarantee that the transportation simplex algorithm will return F^{ID} if there is another optimal flow. In practice, however, the FT iteration always converged to the global minimum of zero in hundreds of randomly generated, perfect-translation examples with $d = L_2$. In these random examples, the points in $\mathbf{y}$ were chosen with coordinates uniformly distributed in $[0, 1]$, and we varied the point space dimension (1, 2, and higher dimensions), whether points all have the same weight or not (if not, then the weight vector u is random), and whether $\Delta y + t^{(0)}$ is the difference between two points in $\mathbf{y}$ or not (if not, then the initial translation $t^{(0)}$ is random).

6.7.4 Equal-Weight Comparisons with Local Minima

In section 6.7.1, we showed examples of unequal-weight comparisons with local minima. These local minima arose because different parts of the heavier distribution were used in an optimal flow for different areas of the transformation space. In this section, we show that there can be local minima even when all the mass in one distribution must be matched to all the mass in the other distribution everywhere in transformation space. This is the matching requirement imposed by the EMD when the distributions are equal-weight.

The main example of this section consists of two distributions over the plane, each having two points ($m = n = 2$). See Figure 6.7(a). We seek to match these distributions under translation using the L_2 and L_1 ground distances.

In section 6.6.3, we analyzed the $m = n = 2$ matching problem under $\mathcal{G}$. Recall that there are two feasible flows F^1 and F^2 such that

$$\begin{array}{ll} F^1 \in \arg\min_{F \in \mathcal{F}(\mathbf{x},\mathbf{y})} \mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y})) & \text{if } d_{11}(g) + d_{22}(g) \le d_{12}(g) + d_{21}(g) \quad \text{and} \\ F^2 \in \arg\min_{F \in \mathcal{F}(\mathbf{x},\mathbf{y})} \mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y})) & \text{if } d_{11}(g) + d_{22}(g) \ge d_{12}(g) + d_{21}(g). \end{array}$$

Here $d_{ij}(g) = d(x_i, g(y_j))$. If $\mathcal{G} = \mathcal{T}$ and $d = L_p$ or $d = L_2^2$, then $d_{ij}(t) = d(x_i, y_j + t) = d(\delta_{ij}, t)$, where $\delta_{ij} = x_i - y_j$.

Now consider the sets

$$\begin{array}{rcll} \mathcal{T}_1 & = & \{\, t \,:\, d_{11}(t) + d_{22}(t) < d_{12}(t) + d_{21}(t) \,\} & \text{and} \\ \mathcal{T}_2 & = & \{\, t \,:\, d_{11}(t) + d_{22}(t) > d_{12}(t) + d_{21}(t) \,\}. & \end{array}$$

If $\mathcal{T}_1 = \emptyset$, then F^2 is optimal for every $t \in \mathcal{T}$; if $T_2 = \emptyset$, then F^1 is optimal for every

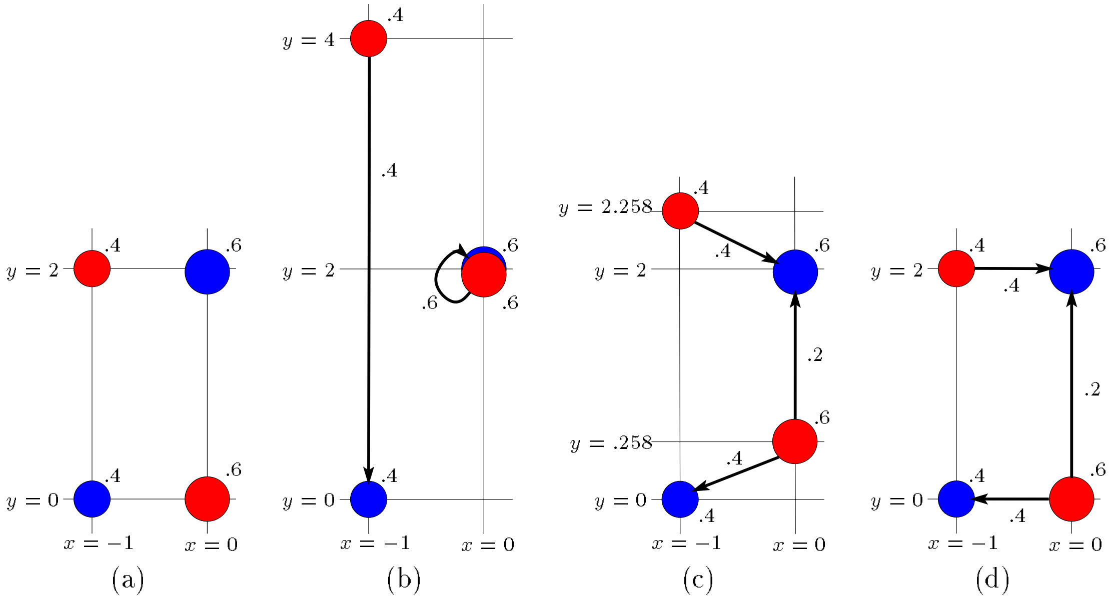

Figure 6.7: A Local Minimum in a 2D Equal-Weight Case. (a) Distributions $\mathbf{x}$ and $\mathbf{y}$ over the plane are shown in blue and red, respectively. (b) The translation $t = [0\ 2]^T$ of $\mathbf{y}$ is locally optimal for both $d = L_2$ and $d = L_1$. The optimal flow for this translation is the same for both ground distances, as is the EMD: $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t) = .6(0) + .4(4) = 1.6$. (c) The globally optimal translation of $\mathbf{y}$ for $d = L_2$ is $t = [0\ .258]^T$. This yields $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t) = .4\sqrt{1^2 + .258^2} + .2(2 - .258) + .4\sqrt{1^2 + .258^2} \doteq 1.175$. (d) The globally optimal translation of $\mathbf{y}$ for $d = L_1$ is $t = [0\ 0]^T$. This yields $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t) = .4(1) + .2(2) + .4(1) = 1.2$.

$t \in \mathcal{T}$. In our example, $x_1 = (-1, 0)$, $x_2 = (0, 2)$, $y_1 = (0, 0)$, $y_2 = (-1, 2)$, $\delta_{11} = [-1\ 0]^T$, $\delta_{12} = [0\ -2]^T$, $\delta_{21} = [0\ 2]^T$, and $\delta_{22} = [1\ 0]^T$. For $d = L_2^2$, it is easy to check that $(d_{11}(t) + d_{22}(t)) - (d_{12}(t) + d_{21}(t)) = -6$ for every $t \in \mathcal{T}$.[9] Thus $\mathcal{T}_2 = \emptyset$, and (as expected) the FT iteration converges to the global minimum on this example with $d = L_2^2$.

In general with $d = L_2$ or $d = L_1$, both $\mathcal{T}_1$ and $\mathcal{T}_2$ will be nonempty. This, however, does not imply that there are local minima. With $d = L_2$ or $d = L_1$, the functions $\text{WORK}(F^1, \mathbf{x}, \mathbf{y} \oplus t)$ and $\text{WORK}(F^2, \mathbf{x}, \mathbf{y} \oplus t)$ are convex in t. If the global minima of these functions occur in $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, then the larger of these values is a local minimum of $\text{WORK}(F, \mathbf{x}, \mathbf{y} \oplus t)$ and the smaller is the global minimum. This is precisely what happens in the example of Figure 6.7(a). A local minimum can also exist along the boundary between $\mathcal{T}_1$ and $\mathcal{T}_2$ where $d_{11}(t) + d_{22}(t) = d_{12}(t) + d_{21}(t)$. More generally, a local minimum may occur in the interior of a transformation space region with constant optimal flow, or along the boundary between two such regions.

Figure 6.7(b) shows $\mathbf{y}$ translated by a locally optimal translation $t = [0\ 2]^T$ for both $d = L_2$ and $d = L_1$, along with the corresponding optimal flow for both cases. The globally optimal translations for the L_2 and L_1 distances are given in Figures 6.7(c) and 6.7(d), respectively. Finally, graphs of $\text{EMD}(\mathbf{x}, \mathbf{y} \oplus t)$ versus t for the L_2 and L_1 distances are shown in Figures 6.8 and 6.9, repectively. In Figure 6.10, we show that the locally optimal and globally optimal translations occur in regions of the translation space for which there are different optimal flows. We also prove that $t = [0\ 2]^T$ is locally optimal for both the L_2 and L_1 ground distances.

Let us now explicitly connect a local minimum in $\delta(g)$ over $\mathcal{G}$ with a local minimum of $\text{WORK}(F, \mathbf{x}, g(\mathbf{y}))$ over $\mathcal{F} \times \mathcal{G}$. Suppose that a local minimum of $\delta(g)$ occurs at g^0 in the interior of a region $R(F^*) = \{\, g \,:\, F^* \in \arg\min_{F \in \mathcal{F}} \text{WORK}(F, \mathbf{x}, g(\mathbf{y})) \,\}$. In words, g^0 is inside the region of transformation space with constant optimal flow F^*. Then there exists a neighborhood $N_\varepsilon^{\mathcal{G}}(g^0) \in \mathcal{G}$ around g^0 of size $\varepsilon > 0$ such that $N_\varepsilon^{\mathcal{G}}(g^0) \subseteq R(F^*)$ and $\delta(g) \geq \delta(g^0)$ for every $g \in N_\varepsilon^{\mathcal{G}}(g^0)$. For every $(F, g) \in \mathcal{F} \times N_\varepsilon^{\mathcal{G}}(g^0)$, we have $\text{WORK}(F, \mathbf{x}, g(\mathbf{y})) \geq \text{WORK}(F^*, \mathbf{x}, g(\mathbf{y})) = \delta(g) \geq \delta(g^0) = \text{WORK}(F^*, \mathbf{x}, g^0(\mathbf{y}))$. The first inequality follows from the optimality of F^* over $N_\varepsilon^{\mathcal{G}}(g^0)$, whereas the second inequality follows from the local optimality of g^0 over $N_\varepsilon^{\mathcal{G}}(g^0)$. Since $\text{WORK}(F, \mathbf{x}, g(\mathbf{y})) \geq \text{WORK}(F^*, \mathbf{x}, g^0(\mathbf{y}))$ for every $(F, g) \in \mathcal{F} \times N_\varepsilon^{\mathcal{G}}(g^0)$, there is a local minimum of $\text{WORK}(F, \mathbf{x}, g(\mathbf{y}))$ at (F^*, g^0). Of course, this logic depends upon being able to fit an open neighborhood inside $R(F^*)$, where F^* is optimal for g^0. This cannot be done, for example, if $R(F^*)$ is the single point g^0. A similar

[9] The fact that this quantity is independent of t is not specific to the particular example of this section. With $d = L_2^2$, $(d_{11}(t) + d_{22}(t)) - (d_{12}(t) + d_{21}(t)) = -2(x_1^T y_1 + x_2^T y_2 - x_1^T y_2 - x_2^T y_1)$ for every $t \in \mathcal{T}$. It follows that there will be one flow which is optimal for every translation.

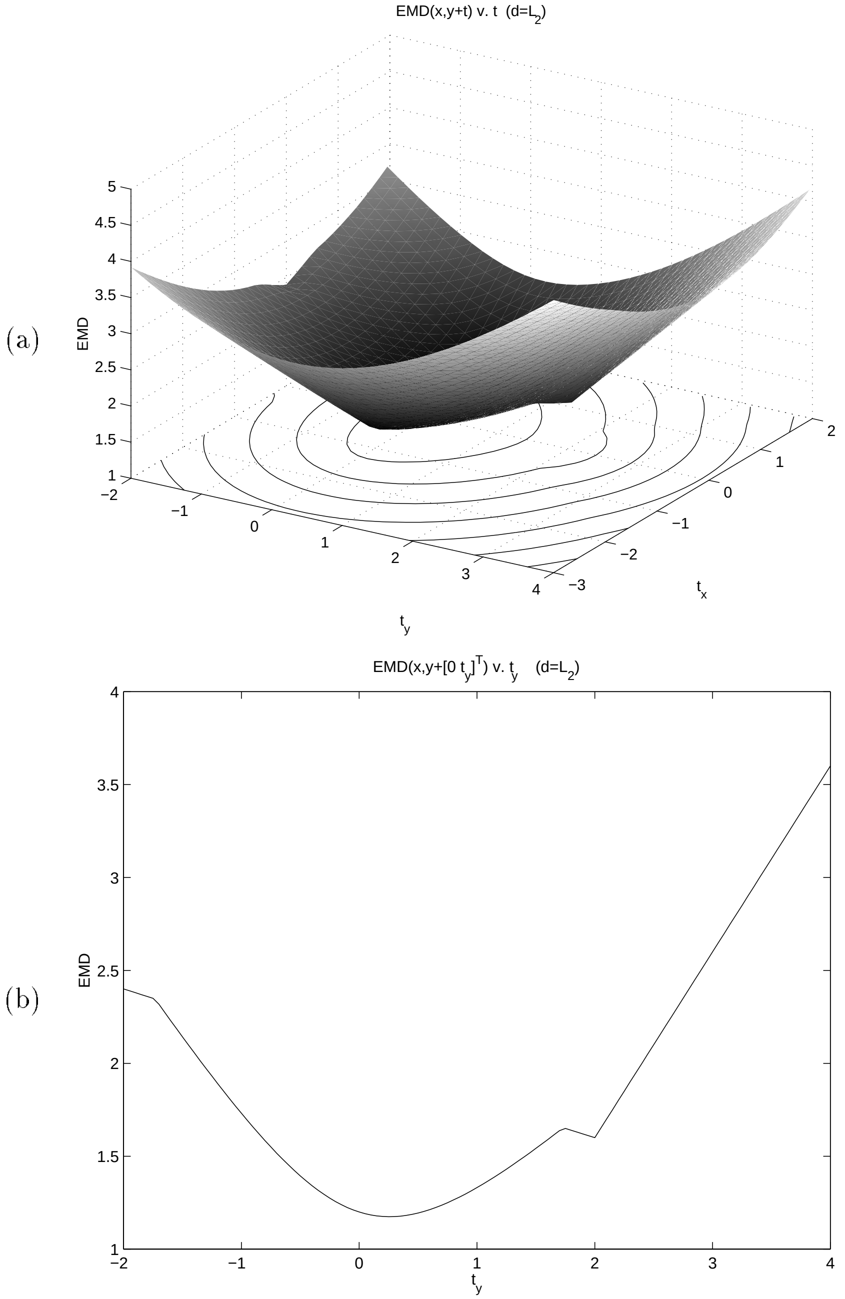

Figure 6.8: Graphs of EMD v. t Showing a Locally Optimal Translation for $d = L_2$. (a) EMD v. t for the 2×2 example shown in Figure 6.7(a) with $d = L_2$. There is a locally optimal translation at $t = [0\ 2]^T$, while the globally optimal translation is $t = [0\ .258]^T$. (b) A slice of the graph in (a) at $t_x = 0$.

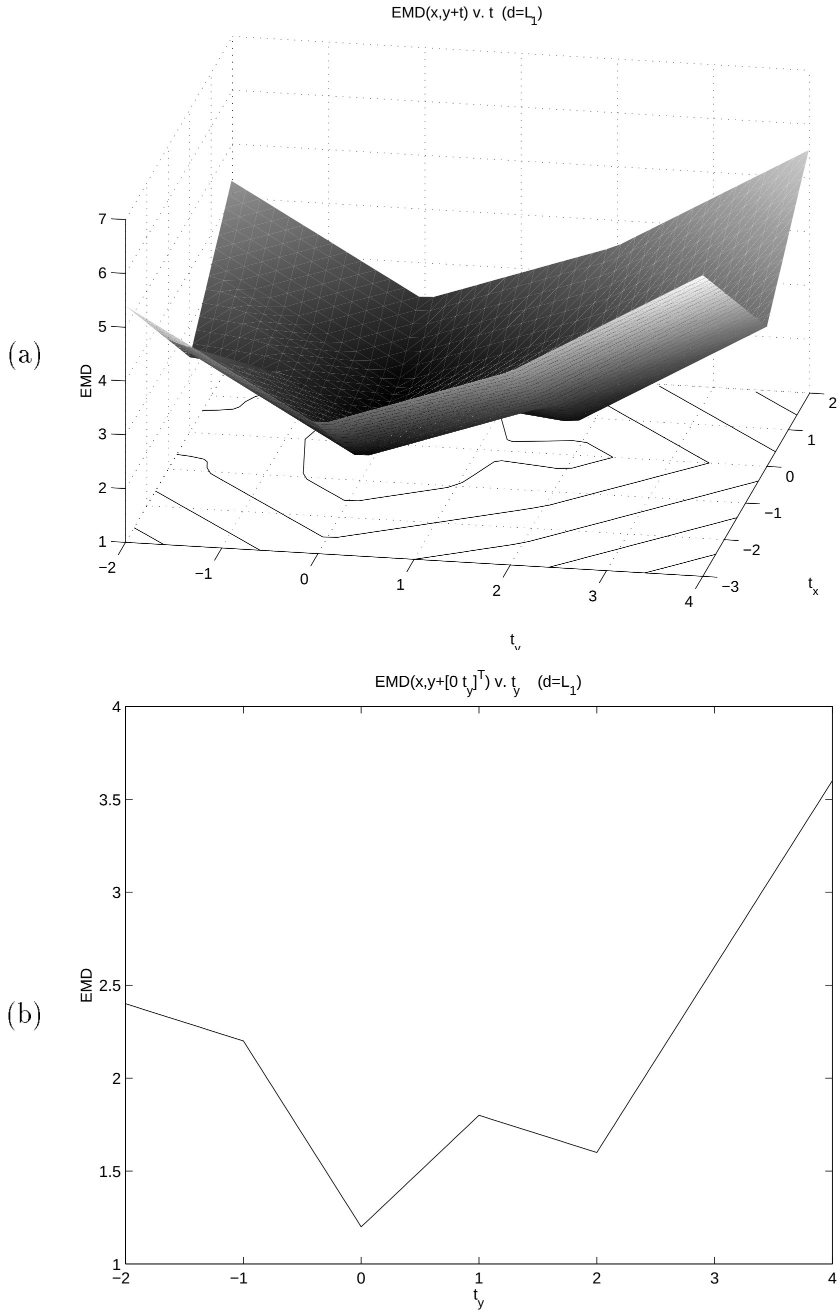

Figure 6.9: Graphs of EMD v. t Showing a Locally Optimal Translation for $d = L_1$. (a) EMD v. t for the 2×2 example shown in Figure 6.7(a) with $d = L_1$. There is a locally optimal translation at $t = [0\ 2]^T$, while the globally optimal translation is $t = [0\ 0]^T$. (b) A slice of the graph in (a) at $t_x = 0$.

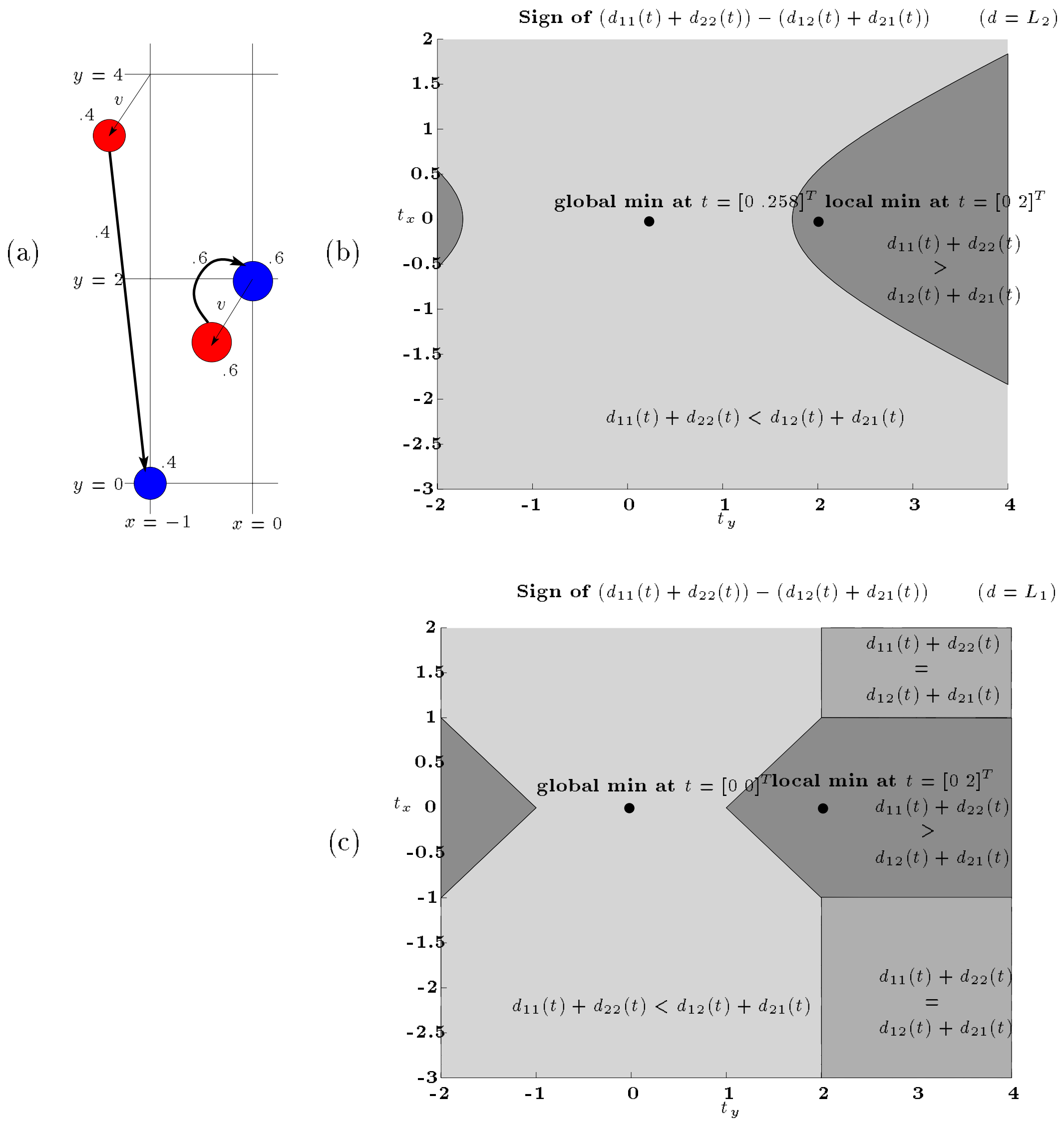

Figure 6.10: A Closer Look at a Locally Optimal Translation. The translation $t = [0\ 2]^T$ is locally optimal in the example shown in Figure 6.7(a) for $d = L_2$ and $d = L_1$. For both these ground distances, $\mathrm{EMD}(\mathbf{x}, \mathbf{y} \oplus t) = 1.6$ (see Figure 6.7(b)). (a) Here we show $\mathbf{y}$ translated by $t + v = [0\ 2]^T + v$, where $||v||$ is small. (b) All translations in the darkest gray area have the same optimizing flow shown in part (a) for $d = L_2$. Here $\mathrm{EMD}(\mathbf{x}, \mathbf{y} \oplus (t + v)) \geq .6||v||_2 + .4(4 - ||v||_2) = .2||v||_2 + 1.6$. Since the EMD is 1.6 when $v = 0$, we see that there is a local minimum at $t = [0\ 2]^T$. The global minimum at $t = [0\ .258]^T$ occurs in an area of translation space where a different flow is optimal. (c) All translations in the darkest gray area have the same optimizing flow shown in part (a) for $d = L_1$. Here $\mathrm{EMD}(\mathbf{x}, \mathbf{y} \oplus (t + v)) \geq .6(|v_x| + |v_y|) + .4(|v_x| + (4 - |v_y|)) = 1.0|v_x| + .2|v_y| + 1.6$. Since the EMD is 1.6 when $v = 0$, we see that there is a local minimum at $t = [0\ 2]^T$. The global minimum at $t = [0\ 0]^T$ occurs in an area of translation space where a different flow is optimal.

connection can be made between a local minimum of $\gamma(F)$ at $F^0 \in \mathcal{F}$ and a local minimum of $\mathrm{WORK}(F, \mathbf{x}, g(\mathbf{y}))$ over $\mathcal{F} \times \mathcal{G}$ if we can fit an open neighborhood inside the set $S(g^*)$ of feasible flows which have g^* as their optimal transformation, where g^* is optimal for F^0.

Of the L_1, L_2, and L_2^2 ground distances, only the L_2^2 distance is guaranteed to yield a WORK function in which a locally optimal translation must be globally optimal. The globally optimal translation for $d = L_2^2$ is the one that lines up the centroids of the two distributions. The centroid of a weighted point set is the point from which the weighted sum of L_2^2 distances to the points in the set is minimized. Recall from section 6.4.1.3 that the spatial and coordinate-wise medians are the points from which the weighted sum of L_2 and L_1 distances, respectively, are minimized. In general, the optimal translation to match two distributions with the EMD, however, is *not* the spatial median for $d = L_2$ and is *not* the coordinate-wise median for $d = L_1$. Indeed, the spatial medians for $\mathbf{x}$ and $\mathbf{y}$ are $(0, 2)$ and $(0, 0)$, respectively, and the coordinate-wise medians are also $(0, 2)$ and $(0, 0)$. In both cases, the locally optimal translation $t = [0\ 2]^T$ lines up the medians, but the globally optimal translation does not.

The magic of the EMD under translation with $d = L_2^2$ does not extend to the EMD under rotation with $d = L_2^2$. In the plane, we seek a rotation angle θ such that $\mathrm{EMD}(\mathbf{x}, R_\theta \mathbf{y})$ is minimized. Even when the L_2^2 ground distance is used, there can be only locally optimal rotation angles. This is clearly shown in Figure 6.11 which contains plots of the EMD versus θ for the example in Figure 6.7(a).

6.8 Odds and Ends

We have not yet discussed the choice of ground distance function used in EMD computations. In section 6.8.1, we consider the tradeoffs in choosing betwen the Euclidean distance and the Euclidean distance squared. One criterion of comparison is solving EMD under transformation problems, although we consider the ground distance choice for other criteria as well. In section 6.8.2, we briefly consider the question: how fast can the EMD between one distribution and a transformed version of another change with respect to the transformation parameters. If the EMD for a given transformation is large, then the EMD for a nearby transformation will also be large if the EMD does not change too quickly. This information may allow a search for an optimal transformation to eliminate a region of the search space without computing the EMD for many transformations in that region.

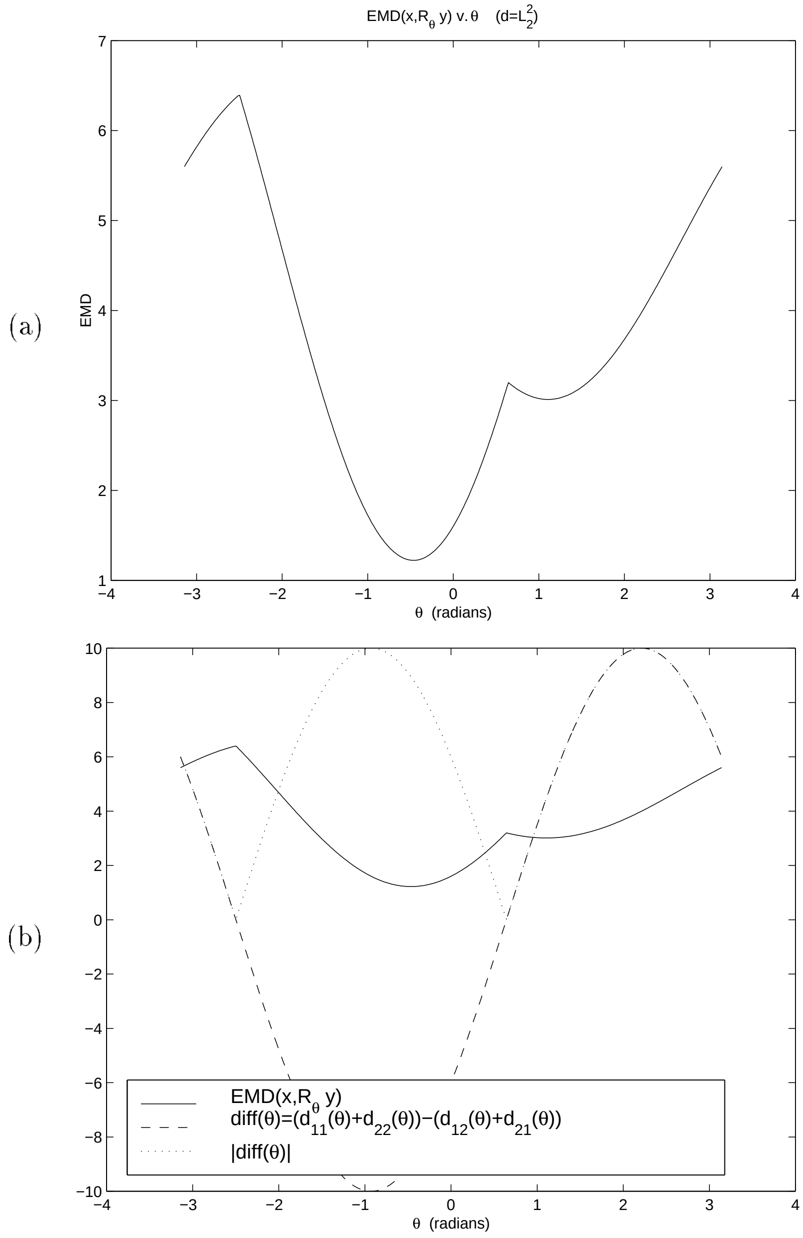

Figure 6.11: Graphs of EMD v. θ Showing a Locally Optimal Rotation for $d = L_2^2$. (a) EMD v. θ for the 2×2 example shown in Figure 6.7(a) with $d = L_2^2$. (b) The globally and locally optimal rotations lie in regions of the rotation space for which there are different optimal flows. The difference function $(d_{11}(\theta) + d_{22}(\theta)) - (d_{12}(\theta) + d_{21}(\theta))$ and its absolute value are the dashed and dotted plots, respectively. The plot of the absolute value shows where the difference function becomes zero and the optimal flow changes.

6.8.1 L_2^2 versus L_2

The EMD takes a "ground distance" function between points and builds upon it a distance function between sets of weighted points. An appropriate ground distance is application-dependent. In the case where the points are located in the CIE-Lab color space, it is natural to use the Euclidean distance as the ground distance since this feature space was specially designed so that perceptual color distance is well-approximated by the L_2 distance. In other feature spaces, the choice may not be so clear. When there is no clear reason to prefer one ground distance over another, it is worth considering the L_2-distance squared even though L_2^2 is not a point metric.

When comparing equal-weight distributions we would like the EMD to be metric so that we cannot have the non-intuitive situation in which two distributions are similar to a third but not to each other. Using an L_p ground distance guarantees that the EMD is a metric between equal-weight distributions. Although this is not the case for L_2^2, the EMD is at most a factor of two away from satisfying the triangle inequality for three given distributions. See section 4.1, formula (4.6).

Another criterion for comparing ground distances is the availability of efficient, effective lower bounds to prune unnecessary EMD computations. In section 5.1.1, we showed that the centroid distance lower bound between equal-weight distributions is valid for both the L_p and L_2^2 ground distances. In section 5.1.2, we used the bound for equal-weight distributions to get a bound on the EMD between unequal-weight distributions. We showed that the minimum ground distance between the centroid of the lighter distribution $\mathbf{y}$ and the centroid of any sub-distribution of $\mathbf{x}$ with the same weight as $\mathbf{y}$ is a lower bound on $\text{EMD}(\mathbf{x}, \mathbf{y})$. Recall that this CLOC lower bound and the more practical CBOX lower bound apply with any ground distance for which the equal-weight centroid bound holds, and this includes $d = L_2^2$ as well as $d = L_2$.

Using L_2^2 also has many advantages in computing the EMD under transformation sets. Consider, for example, the problem of computing the EMD under translation. Applying the FT iteration requires a solution to the optimal translation problem. We gave algorithms for this problem for each of the L_1, L_2 and L_2^2 point distances, but even in the equal-weight case there can be locally optimal translations that are not globally optimal when $d = L_1$ or $d = L_2$ is used. In the equal-weight case with $d = L_2^2$, there are no translations which are only locally optimal. The L_2^2 distance has an even bigger advantage in the FT iteration for higher order transformation sets such as the sets of Euclidean, similarity, linear, and affine transformations. Sum-of-squares optimization problems are well-studied in mathematics, and there are solutions to the optimal transformation problem for each of the listed sets

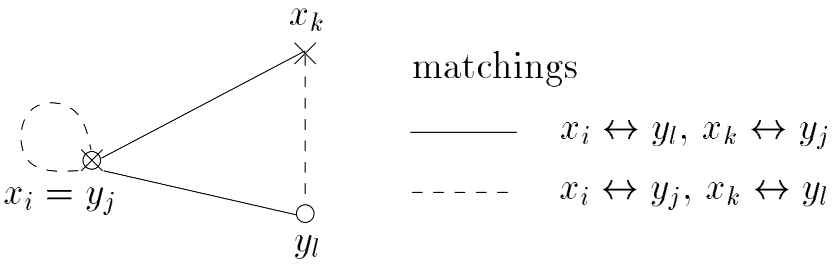
x_k
$x_i = y_j$
y_l
matchings
$x_i \leftrightarrow y_l,\ x_k \leftrightarrow y_j$
$x_i \leftrightarrow y_j,\ x_k \leftrightarrow y_l$

Figure 6.12: Matching Pairs of Points. The two possible matchings of $\{ x_i, x_k \}$ to $\{ y_j, y_l \}$. Here $x_i = y_j$.

(see sections 6.4.2–6.4.3).

The fact that L_2^2 is not a point metric often allows more "natural" optimal flows than for L_p metrics. Consider matching point sets with the EMD. If two points from different sets are on top of each other, then there is always an optimal flow which pairs these two points. This is easily seen with the aid of Figure 6.12. If $d = L_p$, then by the triangle inequality $d(x_i, y_l) + d(x_k, y_j) \geq d(x_k, y_l)$. Thus matching $x_i \leftrightarrow y_l$, $x_k \leftrightarrow y_j$ is at least as expensive as matching $x_i \leftrightarrow y_j$, $x_k \leftrightarrow y_l$. In fact, if $x_i (= y_j)$, x_k, and y_l are not collinear, then the matching with the zero cost correspondence $x_i \leftrightarrow y_j$ is strictly less expensive.

Figure 6.13 gives examples in 1D and 2D to illustrate our previous point. The 1D example in Figure 6.13(a) is due to Jorge Stolfi ([76]). Both flows shown in Figure 6.13(a) are optimal for $d = L_1$ and $d = L_2$, each requiring 6 units of work. The flow on the left costs 36 work units with $d = L_2^2$, while the more natural flow on the right costs 6 work units and is the unique optimal flow for $d = L_2^2$. Figure 6.13(b) shows an example in 2D. The flow on the left is optimal for $d = L_1$ and $d = L_2$, and it is the unique optimal flow since the duplicate point $(0, 0)$ is not involved in a collinearity with two other points from the two sets. The flow on the right is the unique optimal flow for $d = L_2^2$. The two point sets are close to differing by a translation. The correspondences on the right are a lot better for the FT iteration to find the globally optimal translation. Using an L_p ground distance results in "greedy" flows which may give wrong correspondences to the optimal transformation step of the FT iteration.

6.8.2 Sensitivity and Growth Rate

The EMD is insensitive to small perturbations of mass locations. After all, the EMD is a weighted average distance between points, and small changes in point locations result in small changes in inter-point distances. More precisely, we have the following result.

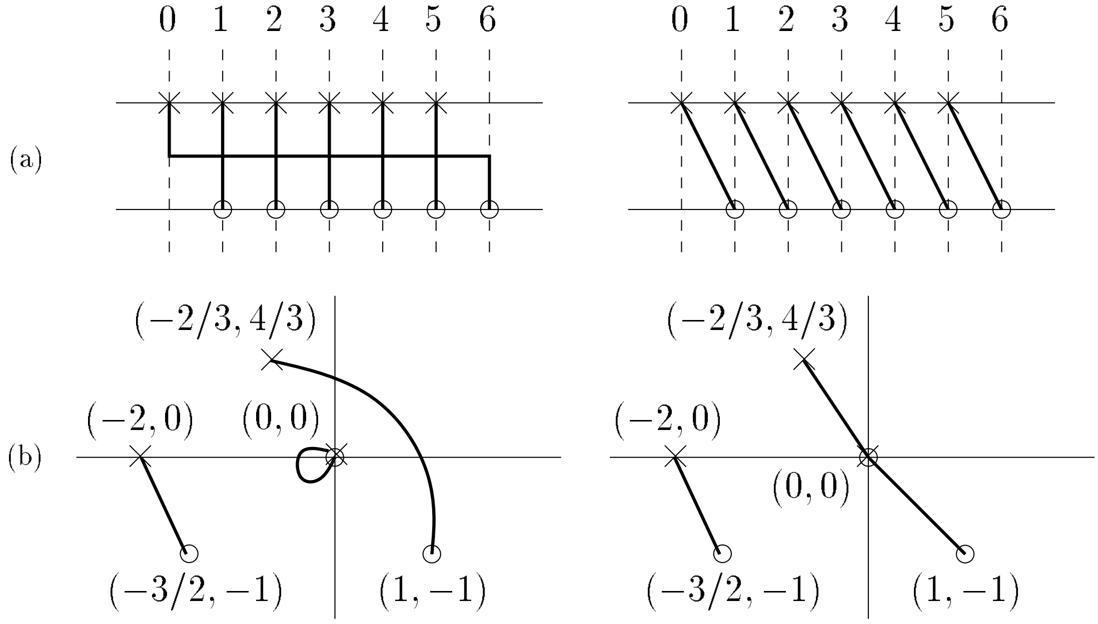

Figure 6.13: Optimal Point Set Matchings under L_2 and L_2^2. The matchings are indicated by dark lines connecting the points. (a) This is a 1D example where the point sets have been offset vertically for clarity. The left and right flows are both optimal for L_2, while the right flow is the unique optimal flow for L_2^2. (b) In this 2D example, the left flow is the unique optimal flow for L_2 and the right flow is the unique optimal flow for L_2^2.

Theorem 14 *If*

$$|d(x_i, g(y_j)) - d(x_i, y_j)| \leq D(g) \qquad \forall i, j, \tag{6.52}$$

then

$$|\mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) - \mathrm{EMD}(\mathbf{x}, \mathbf{y})| \leq D(g). \tag{6.53}$$

Proof. Condition (6.52) implies

$$\begin{aligned} d(x_i, g(y_j)) &\leq d(x_i, y_j) &+& D(g) \quad \forall i, j \quad \text{and} && (6.54)\\ d(x_i, y_j) &\leq d(x_i, g(y_j)) &+& D(g) \quad \forall i, j. && (6.55)\end{aligned}$$

We shall use (6.54) to show that $\mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) - \mathrm{EMD}(\mathbf{x}, \mathbf{y}) \leq D(g)$. Inequality (6.55) implies $\mathrm{EMD}(\mathbf{x}, \mathbf{y}) - \mathrm{EMD}(\mathbf{x}, g(\mathbf{y})) \leq D(g)$ in a completely analogous fashion. Combining these two results yields (6.53).

From (6.54), it follows that

$$\sum_i \sum_j f_{ij} d(x_i, g(y_j)) \leq \sum_i \sum_j f_{ij} d(x_i, y_j) + D(g) \min(w_\Sigma, u_\Sigma) \qquad \forall F \in \mathcal{F}(\mathbf{x}, \mathbf{y}).$$

The left-hand side of the inequality decreases with the replacement of F by an optimal flow $F^*(g) = (f^*_{ij}(g))$ between $\mathbf{x}$ and $g(\mathbf{y})$. Thus

$$\sum_i \sum_j f^*_{ij}(g) d(x_i, g(y_j)) \leq \sum_i \sum_j f_{ij} d(x_i, y_j) + D(g) \min(w_\Sigma, u_\Sigma) \qquad \forall F \in \mathcal{F}(\mathbf{x}, \mathbf{y}).$$

The result follows by dividing both sides by $\min(w_\Sigma, u_\Sigma)$, and replacing F by an optimal flow between $\mathbf{x}$ and $\mathbf{y}$. ■

Note that this result holds regardless of whether or not $\mathbf{x}$ and $\mathbf{y}$ have equal total weight.

For any L_p norm, we have the *reverse triangle inequality* $|\, ||A||_p - ||B||_p \,| \leq ||A - B||_p$.[10] In particular,

$$|\, ||x_i - (y_j + t)||_p - ||x_i - y_j||_p \,| \leq ||t||_p.$$

Thus, Theorem 14 implies[11]

$$\left|\mathrm{EMD}^{||\cdot||_p}(\mathbf{x}, \mathbf{y} \oplus t_1) - \mathrm{EMD}^{||\cdot||_p}(\mathbf{x}, \mathbf{y} \oplus t_2)\right| \leq ||t_1 - t_2||_p. \tag{6.56}$$

[10] Short proof. Apply the triangle inequality twice: (1) $||A||_p - ||B||_p \leq ||A - B||_p$, and (2) $||B||_p - ||A||_p \leq ||B - A||_p = ||A - B||_p$.

[11] The result (6.56) is of the same form (6.53) if we put $\mathbf{z} = \mathbf{y} \oplus t_2$. Then $\mathbf{z} \oplus (t_1 - t_2) = \mathbf{y} \oplus t_1$ and $t = t_1 - t_2$.

In [36], Huttenlocher et al. use an analogous result for the Hausdorff distance to prune translations during the search for a binary model pattern within a binary image. If $\mathrm{EMD}^{||\cdot||_p}(\mathbf{x}, \mathbf{y} \oplus t_1) \geq \alpha$, then $\mathrm{EMD}^{||\cdot||_p}(\mathbf{x}, \mathbf{y} \oplus t_2) \geq \alpha - ||t_1 - t_2||_p$.

Now consider matching planar distributions with $d = L_2$. How fast can $\mathrm{EMD}(\mathbf{x}, R_\theta \mathbf{y} \oplus t)$ change with respect to the Euclidean transformation (R_θ, t)? Here we have

$$|\ ||x_i - (R_\theta y_j + t)||_2 - ||x_i - y_j||_2\ | \leq ||y_j - R_\theta y_j - t||_2 \leq ||y_j - R_\theta y_j||_2 + ||t||_2,$$

where the first and second inequalities follow from the reverse and the ordinary triangle inequalities, respectively. But $||y_j - R_\theta y_j||_2 \leq |\theta|\ ||y_j||_2$ because the length of the arc from y_j to $R_\theta y_j$ is at least the distance between these two points. We can therefore apply Theorem 14 with $D = ||t||_2 + |\theta|(\max_j ||y_j||_2)$. The larger the quantity $\max_j ||y_j||_2$, the weaker the bound (6.53). If we do not replace $||y_j||_2$ by $\max_j ||y_j||_2$ in D, then following the proof of Theorem 14 shows[12]

$$\begin{aligned} &\left|\mathrm{EMD}^{||\cdot||_2}(\mathbf{x}, R_{\theta_1}\mathbf{y} \oplus t_1) - \mathrm{EMD}^{||\cdot||_2}(\mathbf{x}, R_{\theta_2}\mathbf{y} \oplus t_2)\right| \leq \\ &\qquad ||t_1 - t_2||_2 + |\theta_1 - \theta_2| \sum_j (u_j/u_\Sigma)||y_j||_2 \qquad \text{if } u_\Sigma \leq w_\Sigma. \end{aligned}$$

Here we pay an average (instead of worst case) rotational penalty, where each point's norm contribution is weighted by its fraction of the total distribution mass.

6.9 Some Applications

In this section, we apply the FT iteration described in section 6.3 to the problems of illumination-invariant object recognition and point feature matching in stereo image pairs. All experiments were conducted on an SGI Indigo^2 with a 250 MHz processor. The algorithms to solve the transportation problem ([32], pp. 213–229), and the optimal translation, Euclidean, and similarity transformation problems are implemented in C, while the solutions to the optimal linear and affine problems are written in MATLAB.[13]

6.9.1 Lighting-Invariant Object Recognition

Under the assumption of a trichromatic system with a three-dimensional linear model for the surface reflectance functions of object surfaces, Healey and Slater ([28]) showed that

[12] More precisely, use the facts that $|\ ||x_i - (R_{\theta_1} y_j + t_1)||_2 - ||x_i - (R_{\theta_2} y_j + t_2)||_2\ | \leq ||(R_{\theta_2} y_j - R_{\theta_1} y_j) + (t_2 - t_1)||_2 \leq |\theta_2 - \theta_1|\ ||y_j||_2 + ||t_2 - t_1||_2$, and $\sum_i f_{ij} = u_j$ for any feasible flow $F = (f_{ij})$ when $u_\Sigma \leq w_\Sigma$.

[13] Thanks to Yossi Rubner for providing his transportation problem code.

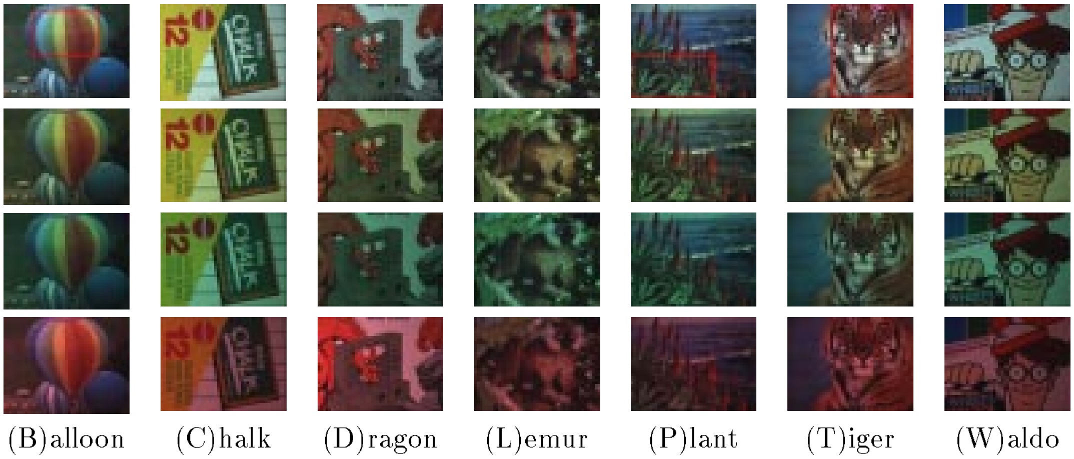

Figure 6.14: Lighting-Invariant Object Recognition – A Small Object Database. An object database imaged under white, yellow, green, and red light is shown in rows 1, 2, 3, and 4, respectively. For some objects, color signatures are computed over only the area outlined in red as shown in row 1. The color signatures for all images of the same object are computed over the same image area, although we only show the red rectangle in the images taken under white light.

an illumination change results in a linear transformation of image pixel colors.[14] In the following experiment, we use a subset of the objects used in [28]. There are four images of each object, one under nearly white illumination and the other three under yellow, green, and red illumination.[15] Figure 6.14 shows the seven database objects imaged under white, yellow, green, and red light.

As in [69], we summarize each image by a set of dominant colors (without regard to position) obtained by clustering in color space, where a color is weighted by the fraction of image pixels classified as that color. We use the clustering algorithm described in [65] in the RGB color space with a minimum bucket size of 16 units in R, G, and B, and we discard clusters with weight less than 0.5%. This produced color signatures with an average of 27 colors.

Our experiment consists of using each image as the query, where the desired distance between images is the EMD under a linear transformation of the corresponding color signatures (with the L_2-distance squared as the ground distance between points in RGB space). To compare a database signature $\mathbf{x}$ to a query signature $\mathbf{y}$, we applied the FT iteration twice (with $\mathcal{G} = \mathcal{L}$): once to transform $\mathbf{y}$ so that it is as close as possible to $\mathbf{x}$, and once

[14] This result also holds for images of scenes with more than one object if all surfaces of all objects have the same basis reflectance functions.

[15] Thanks to David Slater for providing these images.

	B_W	B_Y	B_G	B_R	C_W	C_Y	C_G	C_R	D_W	D_Y	D_G	D_R	L_W	L_Y	L_G	L_R
W	1	3	2	3	1	4	2	3	1	2	3	2	1	4	6	2
Y	3	1	3	5	4	1	3	2	2	1	$\boxed{2}$	4	4	1	4	4
G	2	2	1	2	2	3	1	4	3	3	1	3	3	2	1	3
R	5	5	4	1	3	2	4	1	4	4	4	1	2	3	2	1
Σ	11	11	10	11	10	10	10	10	10	10	10	10	10	10	13	10

	P_W	P_Y	P_G	P_R	T_W	T_Y	T_G	T_R	W_W	W_Y	W_G	W_R
W	1	4	2	3	1	3	2	2	1	4	3	3
Y	4	1	3	2	3	1	3	3	2	1	2	2
G	2	3	1	5	2	2	1	4	4	2	1	4
R	3	2	5	1	4	4	7	1	3	3	4	1
Σ	10	10	11	11	10	10	13	10	10	10	10	10

Figure 6.15: Lighting-Invariant Object Recognition – Query Results. The label at the top of each column shows the query image. The row labels are the illuminants (W)hite, (Y)ellow, (G)reen, and (R)ed. The entry in position (Z, A_X) is the rank of image A_Z in the result for query image A_X. For example, the dragon image for the yellow illuminant is returned as the second closest image when the query image is the dragon image for the green illuminant ($Z = Y$,$A = D$,$X = G$ – see the boxed entry). The number at the bottom of each column is the total of the ranks in that column, where 10 is the ideal value. The query precision is perfect for 21 of the 28 queries, and the average rank sum is 10.4. One run of the FT iteration required an average of 7.4 steps and 4.6 seconds to converge.

to transform $\mathbf{x}$ so that it is as close as possible to $\mathbf{y}$. Both trials were started with the initial transformation equal to the identity map. We use the smaller of these results as the distance between $\mathbf{x}$ and $\mathbf{y}$. The minimum result is equal to $\text{EMD}_{\mathcal{L}}(\mathbf{x}, \mathbf{y})$ if a globally optimal transformation is found, and is greater than $\text{EMD}_{\mathcal{L}}(\mathbf{x}, \mathbf{y})$ otherwise. Ideally, the closest images to the image of an object are the other three images of the same object.

Figure 6.15 shows the results of our experiment.[16] These results are excellent, but not perfect as in [28]. It is possible that we are not finding the globally optimal transformation in some comparisons. Also, the linear transformation model loses accuracy when we replace the color of a pixel by the centroid of a cluster in color space.

6.9.2 Feature Matching in Stereo Images

As we described in section 6.1, the partial EMD under a transformation set can be used to compute the best partial matching of two point sets when one set is free to undergo

[16] The results obtained with all signatures computed over entire images are very similar to the given results.

some transformation.[17] This is exactly the problem we have in matching features extracted from images of the same scene taken from different viewpoints. The fraction parameter γ compensates for the fact that only some features appear in both images, and the set parameter $\mathcal{G}$ accounts for the appropriate transformation between corresponding features. In our experiments, we extract 50 features of a gray level image using an algorithm due to Shi and Tomasi ([74]).[18] The points in the distribution summary of an image are its feature locations (measured in pixels), and the weight of each point is one.[19] The ground distance is the L_2-distance squared between image coordinates. We set $\gamma = 0.5$, so only 25 of the 50 features per image will be matched. Each of the three examples given below uses a different transformation set $\mathcal{G}$, although the initial transformation used in the FT iteration is the identity map in all cases.

In our first example, we match features in two images of a motion sequence in which the camera moves approximately horizontal and parallel to the image plane. Figure 6.16(top) shows the results of applying the FT iteration with $\mathcal{G} = \mathcal{T}$ in an attempt to minimize the partial EMD under a translation of the point features. For this camera motion, all image points translate along the same direction, but the amount of translation for an image point is inversely proportional to the depth of the corresponding scene point ([80]). Thus, the model of a single translation vector is not accurate in general. It is accurate for a set of features that correspond to scene points with roughly the same depth. In this example, the FT iteration matched features on objects toward the back of the table.

The images in the example depicted in Figure 6.16(middle) are also from a motion sequence. Here, however, the camera motion is a forward motion perpendicular to the image plane. The match results shown are the result of applying the FT iteration with $\mathcal{G} = \mathcal{S}$ in an attempt to minimize the partial EMD under a similarity transformation. In our final example, we match features extracted from images of a toy hotel taken from two different viewpoints. Here we apply the FT iteration with $\mathcal{G} = \mathcal{A}$ in an attempt to minimize the partial EMD under an affine transformation of feature locations. The match results are shown in Figure 6.16(bottom). In all three cases, it appears that the FT iteration converged to a globally optimal transformation. In many examples, however, running the iteration once leads to only a locally optimal solution.

[17] Recall that the same code used to solve the transportation problem can be used to solve the assignment problem and to compute the partial EMD.

[18] Thanks to Stan Birchfield for his implementation of this feature extraction algorithm.

[19] Using the gray levels in a small area around a feature in addition to its location may improve matching results. However, corresponding pixels in images of a scene from different viewpoints may have gray level differences which are not small. Therefore, using gray level information may hurt matching results if we do not account for such differences.

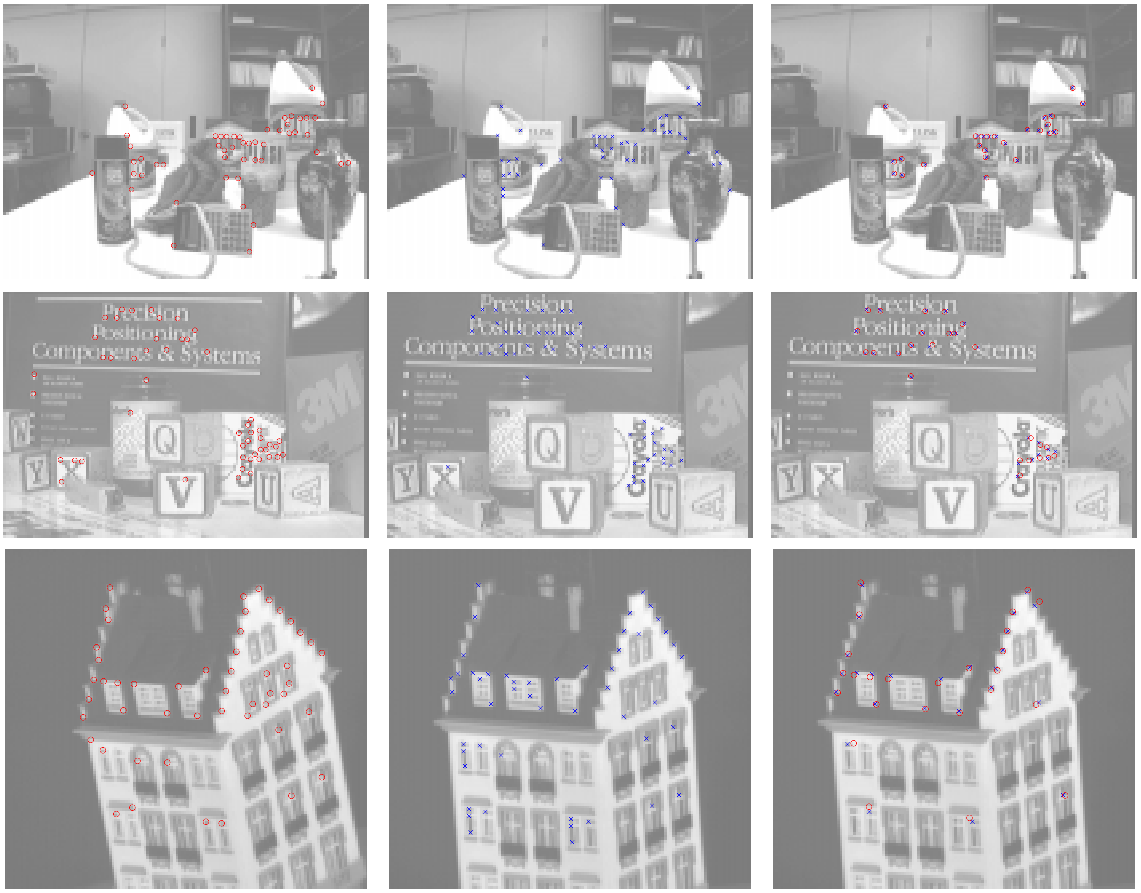

Figure 6.16: Point Feature Matching in Stereo Images – Results. The first two columns in each row show two images and the locations of 50 features in each image. The last column shows the result of matching the features using the FT iteration with an initial transformation equal to the identity map. Here $\gamma = 0.5$, so only 25 features are matched in each example. We report the number of steps S and the time T in seconds (s) for the FT iteration to converge. (top) $\mathcal{G} = \mathcal{T}$, $S = 11$, $T = 1.77$s. (middle) $\mathcal{G} = \mathcal{S}$, $S = 4$, $T = 1.08$s. (bottom) $\mathcal{G} = \mathcal{A}$, $S = 8$, $T = 36.21$s.