



Published in final edited form as:

*Int J Comput Vis.* 2013 January 1; 101(2): 254–269. doi:10.1007/s11263-012-0566-z.

## A linear optimal transportation framework for quantifying and visualizing variations in sets of images

**Wei Wang,**

Center for Bioimage Informatics, Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

**Dejan Slepčev,**

Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

**Saurav Basu,**

Center for Bioimage Informatics, Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

**John A. Ozolek,** and

Department of Pathology, Children's Hospital of Pittsburgh, Pittsburgh, PA, 15224 USA

**Gustavo K. Rohde**

Center for Bioimage Informatics, Department of Biomedical Engineering, Department of Electrical and Computer Engineering, Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA, 15213 USA. Tel.: +1-412-268-3684, Fax: +1-412-268-9580

Wei Wang: [wwang2@andrew.cmu.edu](mailto:wwang2@andrew.cmu.edu); Dejan Slepčev: [slepcev@math.cmu.edu](mailto:slepcev@math.cmu.edu); Saurav Basu: [sauravb@andrew.cmu.edu](mailto:sauravb@andrew.cmu.edu); John A. Ozolek: [ozolekja@upmc.edu](mailto:ozolekja@upmc.edu); Gustavo K. Rohde: [gustavor@cmu.edu](mailto:gustavor@cmu.edu)

### Abstract

Transportation-based metrics for comparing images have long been applied to analyze images, especially where one can interpret the pixel intensities (or derived quantities) as a distribution of 'mass' that can be transported without strict geometric constraints. Here we describe a new transportation-based framework for analyzing sets of images. More specifically, we describe a new transportation-related distance between pairs of images, which we denote as linear optimal transportation (LOT). The LOT can be used directly on pixel intensities, and is based on a linearized version of the Kantorovich-Wasserstein metric (an optimal transportation distance, as is the earth mover's distance). The new framework is especially well suited for computing all pairwise distances for a large database of images efficiently, and thus it can be used for pattern recognition in sets of images. In addition, the new LOT framework also allows for an isometric linear embedding, greatly facilitating the ability to visualize discriminant information in different classes of images. We demonstrate the application of the framework to several tasks such as discriminating nuclear chromatin patterns in cancer cells, decoding differences in facial expressions, galaxy morphologies, as well as sub cellular protein distributions.

### Keywords

Optimal transportation; linear embedding

# 1 Introduction

## 1.1 Background and motivation

Automated image analysis methods are often used for extracting important information from image databases. Common applications include mining information in sets of microscopy images [22], understanding mass distribution in celestial objects from telescopic images [34], as well as analysis of facial expressions [36], for example. We note that the prevalent technique for quantifying information in such large datasets has been to reduce the entire information (described by pixel intensity values) contained in each image in the database to a numerical feature vector (e.g. size, form factor, etc.). Presently, this approach (coupled with clustering, classification, and other machine learning techniques) remains the prevalent method through which researchers extract quantitative information about image databases and cluster important image subgroups [6,44,19].

In addition to feature-based methods, several techniques for analyzing image databases based on explicit modeling approaches have recently emerged. Examples include contour-based models [29], medial axis models [8], as well as model-based deconvolution methods [15]. Moreover, when analyzing images within a particular type (e.g. brain images) researchers have often used a more geometric approach. Here the entire morphological exemplar as depicted in an image is viewed as a point in a suitably constructed metric space (see for example [4]), often facilitating visualization. These approaches have been used to characterize the statistical variation of a particular object in a given population (or set of populations) [4,24]. The main idea in these is to understand the variation of similar objects through analysis of the deformation fields required to warp one object (as depicted in its image) onto another.

Alternatively, transportation-based metrics have also been used to analyze image data in problems when pixel intensities (or derived quantities) can be interpreted as ‘mass’ free to move without strict geometric constraints [32,18,11,16,43]. They are interesting alternatives to other methods since when applied directly to pixel intensities, they have the potential to quantify both texture and, to some extent, shape information combined (see for example [18,43]). In particular, there has been continuing effort to develop fast and reliable methods for computing transportation related distances [26,20, 35,28,2,3,5,12,17]. While the computational complexity of these methods ranges from quadratic to linear with respect to image size (for smooth enough images), the computations are still expensive (time wise), in particular for rough images (with large gradients), and there are issues with convergence (e.g. due to local minima in PDE-based variational implementations).

## 1.2 Overview of our approach and contributions

Our contribution is to develop a linear framework closely related to the optimal transportation metric (OT) [43] for analyzing sets of images. This framework, which we call the linear optimal transportation (LOT), not only provides a fast way for computing a metric (and geodesics) between all pairs in a dataset, permitting one to use pixel intensities directly, but it also provides an isometric embedding for a set of images. Therefore *our method takes as input a potentially large set of images and outputs an isometric embedding of the dataset (endowed with the LOT metric) onto the standard Euclidean space*. Our approach achieves this task by utilizing the following series of steps:

- **Step 1:** compute a template image that will serve as a reference point for analyzing the given image dataset.

- **Step 2:** for each image in the input dataset, as well as the estimated template, compute a particle approximation that will enable a linear programming-based computation of the OT distance between it and the template computed in STEP 1.
- **Step 3:** normalize each particle approximation with respect to translation, rotation, and coordinate inversions.
- **Step 4:** compute a quadratic-based OT distance between the particle approximation of each image and the template.
- **Step 5:** from the output of STEP 4, compute the LOT distances and embedding.

As compared to previous works that make use of transportation-related metrics in image analysis, we highlight the following innovations of our approach. The first is that, given a database of  $M$  images, the number of transportation related optimizations required for computing the distance between all pairs of images is  $M$  when utilizing our approach (versus  $M(M-1)/2$  when utilizing other approaches). This provides a substantial increase in speed especially when performing pattern recognition tasks (e.g. classification) in large databases. Secondly, as mentioned above, our LOT framework also provides an isometric linear embedding that has a couple of convenient properties. One being that the embedding of an image newly added to the database can be computed exactly and with one transportation optimization only. Another being that any point in the embedded space can be visualized as an image. This includes measured points (existing images in the database) but also any other point in this space. Below we show how this embedding greatly facilitates the use of standard geometric data analysis techniques such as principal component analysis (PCA) and linear discriminant analysis (LDA) for visualizing interesting variations in a set of images.

### 1.3 Paper organization

In section 2 we begin by reviewing the mathematical underpinnings of the traditional optimal transportation framework and then describe its linearized version and some of its properties. Equation (1) gives the definition of the traditional OT metric, and formulas (2) and (3) the linearized version of the metric. Then in section 3 we describe our computational approach in the discrete setting. This includes an algorithm for ‘estimating’ the content of a relatively sparse image with particles (full details of this algorithm are provided in appendix A). The definition of the OT distance in the discrete setting is provided in equation (5), while our approximation of its linearized version in the discrete setting is given in formula (9). Equations (10) and (11) provide the isometric linear embedding for a given particle approximation according to a reference point. Section 4 describes how one can utilize the linear embedding provided in the LOT framework to extract useful information from sets of images using principal component analysis and linear discriminant analysis. The applications of LOT towards decoding subcellular protein patterns and organelle morphology, galaxy morphology as well as facial expression differences are presented in section 5.

## 2 Optimal transportation framework

Transportation-based metrics are especially suited for quantifying differences between structures (depicted in quantitative images) that can be interpreted as a distribution of mass with few strict geometric constraints. More precisely, we utilize the optimal transportation (Kantorovich-Wasserstein) framework to quantify how much mass, in relative terms, is distributed in different regions of the images. We begin by describing the mathematics of the traditional OT framework, and in particular the geometry behind it, which is crucial to our subsequent introduction of the linearized OT distance.

### 2.1 Optimal transportation metric

Let  $\Omega$  represent the domain (the unit square  $[0, 1]^2$ , for example) over which images are defined. To describe images we use the mathematical notion of a measure. While this is somewhat abstract, it enables us to treat simultaneously the situation when we consider the image to be a continuous function and when we deal with actual computations and consider the image as a discrete array of pixels. It is important for us to do so, because many notions related to optimal transportation are simpler in the continuous setting, and in particular we can give an intuitive explanation for the LOT distance we introduce. On the other hand the discrete approximation necessitates considering a more general setting because the mass (proportional to intensity) from one pixel during transport often needs to be redistributed over several pixels.

We note that, in the current version of the technique, we normalize all images in a given dataset so that the intensity of all pixels in each image sums to one. Thus we may interpret images as probability measures. The assumption is adequate for the purpose of analyzing shape and texture in the datasets analyzed in this paper. We note that OT-related distances can be used when masses are not the same [32,27].

Recall that probability measures are nonnegative and that the measure of the whole set  $\Omega$  is 1:  $\mu(\Omega) = \nu(\Omega) = 1$ . Let  $c: \Omega \times \Omega \rightarrow [0, \infty)$  be the *cost function*. That is  $c(x, y)$  is the ‘cost’ of transporting unit mass located at  $x$  to the location  $y$ . The optimal transportation distance measures the least possible total cost of transporting all of the mass from  $\mu$  to  $\nu$ . To make this precise, consider  $\Pi(\mu, \nu)$ , the set of all *couplings* between  $\mu$  and  $\nu$ . That is consider the set of all probability measures on  $\Omega \times \Omega$  with the first marginal  $\mu$  and the second marginal  $\nu$ . More precisely, if  $\pi \in \Pi(\mu, \nu)$  then for any measurable set  $A \subset \Omega$  we have  $\pi(A \times \Omega) = \mu(A)$  and  $\pi(\Omega \times A) = \nu(A)$ . Each coupling describes a *transportation plan*, that is  $\pi(A_0 \times A_1)$  is telling one how much ‘mass’ originally in the set  $A_0$  is being transported into the set  $A_1$ .

We consider optimal transportation with quadratic cost  $c(x, y) = |x - y|^2$ . The optimal transportation (OT) distance, also known as the Kantorovich-Wasserstein distance, is then defined by

$$d_w(\mu, \nu) = \left( \inf_{\pi \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} |x - y|^2 d\pi \right)^{\frac{1}{2}}. \quad (1)$$

It is well known that the above infimum is attained and that the distance defined is indeed a metric (satisfying the positivity, the symmetry, and the triangle inequality requirements), see [39]. We denote the set of minimizers  $\pi$  above by  $\Pi_{OT}(\mu, \nu)$ .

### 2.2 Geometry of optimal transportation in continuous setting

The construction of the LOT metric which we introduce below, can be best motivated in the continuous setting. Consider measures  $\mu$  and  $\nu$  which have densities  $\alpha$  and  $\beta$ , that is

$$d\mu = \alpha(x)dx \text{ and } d\nu = \beta(x)dx.$$

Then the following mathematical facts, available in [39], hold. The OT plan between  $\mu$  and  $\nu$  is unique and furthermore the mass from each point  $x$ , is sent to a single location, given as the value of the function  $\phi(x)$  called the *optimal transportation map* (see Figure 1). The relation between  $\phi$  and the optimal transportation plan  $\Pi$  introduced above is that  $\Pi(A_0 \times A_1) = \mu(\{x \in A_0 : \phi(x) \in A_1\})$ . That is  $\Pi$  is concentrated on the graph of  $\phi$ . We note that  $\phi$  is

a measure preserving map from  $\mu$  to  $\nu$ , that is that for any  $A$ ,  $\int_{\phi^{-1}(A)} \alpha(x) dx = \int_A \beta(y) dy$ . The Kantorovich-Wasserstein distance is then  $d_W(\mu, \nu) = \int_{\Omega} |\phi(x) - x|^2 \alpha(x) dx$ .

In addition to being a metric space the set of measures is formally a *Riemannian manifold* [10], that is at any point there is a tangent space endowed with an inner product. In particular the tangent space at the measure  $\sigma$  with density  $\gamma$  (i.e.  $d\sigma = \gamma(x) dx$ ) is the set of the following vector fields  $T_{\sigma} = \{v : \Omega \rightarrow \mathbb{R}^d \text{ such that } \int_{\Omega} |v(x)|^2 \gamma(x) dx < \infty\}$  and the inner product is the weighted  $L^2$ :

$$\langle v_1, v_2 \rangle_{\sigma} = \int_{\Omega} v_1(x) \cdot v_2(x) \gamma(x) dx.$$

The OT distance is just the length of the shortest curve (geodesic) connecting two measures [5].

A very useful fact is the *geodesics* have a form that is simple to understand. Namely if  $\mu_t$ ,  $0 \leq t \leq 1$  is the geodesic connecting  $\mu$  to  $\nu$ . Then  $\mu_t$  is the measure obtained when mass from  $\mu$  is transported by the transportation map  $x \rightarrow (1-t)x + t\phi(x)$ . Then  $\mu_t(A) = \int_{\phi_t^{-1}(A)} \alpha(x) dx$ .

### 2.3 Linear optimal transportation metric

Computing a pairwise OT distance matrix is expensive for large datasets. In particular let  $T_{OT}$  be the time it takes to compute the OT distance and  $T_2$  the time it takes to compute the Euclidean distance between two moderately complex images. Using the method described below,  $T_{OT}$  is typically on the order of tens of seconds, while  $T_2$  is on the order of milliseconds. For a dataset with  $M$  images, computing the pairwise distances takes time on the order of  $M(M-1)T_{OT}/2$ . Here we introduce a version of the OT distance that is much faster to compute. In particular computing the distance matrix takes approximately  $MT_{OT} + M(M-1)T_2/2$ . For a large set of images, in which the number of pixels in each image is fixed, as the number of images in the set tends to infinity, the dominant term is  $M^2 T_2$ .

The distance we compute is motivated by the geometric nature of the OT distance. Heuristically, instead of computing the geodesic distance on the manifold we compute a ‘projection’ of the manifold to the tangent plane at a fixed point and then compute the distances on the tangent plane. This is the main reason we use the term *linear* when naming the distance. To consider the projection one needs to fix a reference image  $\sigma$  (where the tangent plane is set). Computing the projection for each image requires computing the OT plan between the reference image  $\sigma$  and the given image. Once computed, the projection provides a natural linear embedding of the dataset, which we describe in section 3.5.

We first describe the LOT distance in the continuous setting. Let  $\sigma$  be a probability measure with density  $\gamma$ . We introduce the identification (‘projection’),  $P$ , of the manifold with the tangent space at  $\sigma$ . Given a measure  $\mu$ , consider the optimal transportation map  $\psi$  between  $\sigma$  and  $\mu$ . Then

$$P(\mu) = v \text{ where } v(x) = \psi(x) - x.$$

Note that  $v \in T_{\sigma}$  and that  $P$  is a mapping from the manifold to the tangent space. See Figure 2 for a visualization. Also  $P(\sigma) = 0$  and

$$d_w(\sigma, \mu)^2 = \int_{\Omega} |\psi(x) - x|^2 \gamma(x) dx = \int_{\Omega} |v(x)|^2 \gamma(x) dx = \langle v, v \rangle_{\sigma} = \|P(\mu) - P(\sigma)\|_{\sigma}^2, \text{ where } \|v\|_{\sigma}^2 \text{ is defined to be } \langle v, v \rangle_{\sigma}. \text{ So the mapping preserves distances to } \sigma. \text{ Let us mention that in cartography}$$

such projection is known as the equidistant azimuthal projection. In differential geometry this would be the inverse of the exponential map. We define the LOT as:

$$d_{LOT}(\mu, \nu) = \|P(\mu) - P(\nu)\|_{\sigma}. \quad (2)$$

When  $\sigma$  is not absolutely continuous with respect to the Lebesgue measure the situation is more complicated. The purely discrete setting (when the measures are made of particles) is the one relevant for computations and we discuss it in detail in section 3.3. In the general setting, the proper extension of the LOT distance above is the shortest *generalized geodesic* (as defined in [1]) connecting  $\mu$  and  $\nu$ . More precisely, given a reference measure  $\sigma$ , and measures  $\mu$  and  $\nu$ , with transportation plans  $\pi_{\mu} \in \Pi_{OT}(\sigma, \mu)$  and  $\pi_{\nu} \in \Pi_{OT}(\sigma, \nu)$ , let  $\Pi(\sigma, \mu, \nu)$  be the set of all measures on the product  $\Omega \times \Omega \times \Omega$  such that the projection to first two co-ordinates is  $\pi_{\mu}$  and the projection to the first and the third is  $\pi_{\nu}$ . The linearized version of the OT distance between  $\mu$  and  $\nu$  is given by:

$$d_{LOT,\sigma}(\mu, \nu)^2 = \inf_{\pi \in \Pi(\sigma, \mu, \nu)} \int_{\Omega \times \Omega \times \Omega} |x - y|^2 d\pi. \quad (3)$$

### 2.4 Translation and Rotation normalization

It is important to note that the OT metric defined in equation (1) is not invariant under translations or rotations. It can be rendered translation invariant by simply aligning all measures in a dataset  $\mu_1, \dots, \mu_N$  by setting their center of mass to a common coordinate. This is based on the fact that if  $\mu$  is a measure on  $\Omega$  with center of mass

$$x_{\mu} = \frac{1}{\mu(\Omega)} \int_{\Omega} x d\mu$$

and  $\nu$  a measure of the same mass, then among all translates of the measure  $\nu$ ,  $\nu_x(A) = \nu(A - x)$ , the one that minimizes the distance to  $\mu$  is the one with the center of mass  $x_{\mu}$ .

From now on, we will always assume that the measures are centered at the origin (the implementation of this normalization step is given below). The Euclidean-transformation invariant Kantorovich-Wasserstein distance is then defined in the following way. Given a measure  $\mu$  and an orthogonal matrix  $T$ , define  $\mu_T$  by  $\mu_T(A) = \mu(T^{-1}A)$ . Then the invariant distance is defined by  $d(\mu, \nu) = \min_{T \in O(n)} d_W(\mu_T, \nu)$ . In two dimensions we have developed an algorithm for finding the minimum above. However we know of no reasonable algorithm for computing the rotation invariant linearized OT distance ( $d_{LOT}$ ). Below we present an algorithm that greatly reduces the effect of rotation, but does not eliminate it completely.

### 3 Computing LOT distances from image data

To compute the LOT distance for a set of images we need to first compute the OT distance from a template image to each of the images. There exist several approaches for computing OT distances. Perhaps the most direct idea is to discretize the problem defined in (1). This results in a linear programming problem (see (5) below). While this approach is very robust and leads to the global minimizer, it is computationally expensive (the fastest strongly polynomial algorithm for this case is the Hungarian algorithm with time complexity of  $O(n^3)$  with  $n$  the number of pixels in an image). Other approaches are based on continuum ideas, where PDEs and variational techniques are often used [2,3,5,12,17]. Some of them achieve

close to linear scaling with the number of pixels. On the other hand, their performance can deteriorate as one considers images that lack regularity (are not smooth, have large gradients). A particular difficulty with some of the approaches that are not based on linear programming is that they may not converge to the global solution if they arrive at a local minimum first.

Our approach is based on combining linear programming with a particle approximation of the images. It is a refinement of the algorithm we used in [43]. That is, each image is first carefully approximated by a particle measure (a convex combination of delta masses) that has much fewer particles than there are pixels in the image. This significantly reduces the complexity of the problem. Furthermore it takes advantage of the fact that many images we consider are relatively sparse. Then one needs fewer particles for accurate approximation of the image which accelerates the computation. In addition, as the number of particles approaches the number of pixels in the image, the approximation error tends to zero. We now present the details of the algorithm. In particular, we give a description of the particle approximation, and detailed algorithms for translation and rotation normalization, and the computation of the LOT distance. The full details pertaining to the particle approximation algorithm, which is more involved, are presented in Appendix A.

### 3.1 Particle approximation

The first step in our computational approach is to represent each image as a weighted combination of ‘particles’ each with mass (intensity)  $m_i$  and location  $x_i$ :

$$\mu = \sum_{i=1}^{N_\mu} m_i \delta_{x_i}. \quad (4)$$

with  $N_\mu$  being the number of masses used to represent the measure (image)  $\mu$ . Since the images are digital, each image could be represented exactly using the model above by selecting  $m_i$  to be the pixel intensity values and  $\delta_{x_i}$  the pixel coordinate grid. Given that images can contain a potentially large number of pixels, and that the cost of a linear programming solution for computing the OT is generally  $O(N_\mu^3)$ , we do not represent each image exactly and instead choose to approximate them.

The goal of the algorithm is to use at most, approximately,  $N$  particles to approximate each image, while maintaining a similar approximation error (by approximation error we mean the OT distance between the image and the particle approximation) for all images of a given dataset. For a dataset of  $M$  images, our algorithm requires the user to specify a number  $N$  for the approximate number of particles that could be used to approximate any image, and consists of these four steps:

- **Step 1:** use a weighted  $K$ -means algorithm [21] to approximate each image, with the number of clusters set to the chosen  $N$ . As we describe in the appendix, the weighted  $K$ -means is an optimal local step for reducing the OT error between a given particle approximation and an image.
- **Step 2:** improve the approximation of each image by investigating regions where each image is not approximated well, and add particles to those regions to improve the approximation locally. Details are provided in Appendix A.
- **Step 3:** compute the OT distance between each digital image and the approximation of each image. For images not well approximated by the current algorithm (details provided in Appendix A), add particles to improve the approximation.

- **Step 4:** For images where the approximation error is significantly smaller than the average error (for the entire dataset) remove (merge) particles until the error between the digital image and its approximation is more similar to the other images. (This reduces the time to compute the distances, while it does not increase the typical error.)

The algorithm above was chosen based on mathematical observations pertaining to the OT metric. These observations, including a careful description of each of the steps above, are provided in Appendix A. We have experimented with several variations of the steps above, and through this experience we have converged on the algorithm presented. A few of the details of the algorithm could be improved further, though we view further improvements to be beyond the scope of this paper. We emphasize that this particle approximation algorithm works particularly well for relatively sparse images, and less so for non-sparse (e.g. flat) images. Below we demonstrate, however, that the algorithm can nonetheless be used to extract meaningful quantitative information for datasets of images which are not sparse.

**Translation and rotation normalization**—In two dimensions, it suffices to consider translation, rotation, and mirror symmetry. To achieve that we simply align the center of mass of each particle approximation to a fixed coordinate, and rotate each particle approximation to a standard reference frame according to a principal axis (Hotteling) transform. Each particle approximation is then flipped left to right, and up and down, simply by reversing their coordinates, until the skewness (the third standardized moment) of the coordinates of each sample have the same sign.

### 3.2 OT distance in particle setting

We start by explaining how the general framework of section 2 applies to particle measures (such as particle approximations obtained above). A particle probability measure,  $\mu$ , which approximates the image is given as  $\mu = \sum_{i=1}^N m_i \delta_{x_i}$ , where  $x_i \in \Omega$ ,  $m_i \in (0, 1]$ ,  $\sum_{i=1}^N m_i = 1$ .

We note that for  $A \subset \Omega$ ,  $\mu(A) = \sum_{i: x_i \in A} m_i$ . An integral with respect to measure  $\mu$  is  $\int_A f(x) d\mu(x) = \sum_{i: x_i \in A} f(x_i) m_i$ .

Now we turn to our main goal of the section: obtaining the OT distance. For  $\mu = \sum_{i=1}^{N_\mu} m_i \delta_{x_i}$  and  $\nu = \sum_{j=1}^{N_\nu} p_j \delta_{y_j}$  the set of *couplings*  $\Pi(\mu, \nu)$  is given by a set of  $N_\mu \times N_\nu$  matrices, as follows:

$$\Pi(\mu, \nu) = \left\{ \sum_{i=1}^{N_\mu} \sum_{j=1}^{N_\nu} f_{i,j} \delta_{x_i, y_j}; f_{i,j} \geq 0 \text{ for } i=1, \dots, N_\mu, j=1, \dots, N_\nu \sum_{j=1}^{N_\nu} f_{i,j} = m_i \text{ for } i=1, \dots, N_\mu \sum_{i=1}^{N_\mu} f_{i,j} = p_j \text{ for } j=1, \dots, N_\nu \right\}.$$

Since it is clear from the context we will make no distinction between measures in  $\Pi(\mu, \nu)$  and matrices  $f = [f_{i,j}]$  that satisfy the conditions above.

The optimal transportation distance between  $\mu$  and  $\nu$  defined in (1) is the solution to the following linear programming problem:

$$d_w^2(\mu, \nu) = \min_{f \in \Pi(\mu, \nu)} \sum_{i=1}^{N_\mu} \sum_{j=1}^{N_\nu} |x_i - y_j|^2 f_{i,j}. \quad (5)$$



See Figure 3 for a visualization. For convenience, we utilize Matlab’s implementation of a variation of Mehrotra’s dual interior point method [23] to solve the linear program. We note however that better suited alternatives, which take advantage of the special structure of the linear programming problem, exist and could be used [26,32] to increase the computational efficiency of the method even further. The solution of the linear program above is relatively expensive [43], with typical computation times being around 30 seconds (with 500 particles per image) on an Apple laptop with a 2.2GHz intel dual core processor and 2GB of RAM.

For datasets containing tens of thousands of images this is practical only if we need to compute a single OT distance per image (the one to the reference image), and is prohibitive if we need the pairwise distance between all images. This is the reason we introduce the linear optimal transportation framework to compute a related distance between  $\mu$  and  $\nu$ .

### 3.3 LOT distance in particle setting

Computing the linear optimal transportation (LOT) distance requires setting a reference

template  $\sigma$ , which we also choose to be a particle measure  $\sigma = \sum_{k=1}^{N_\sigma} q_k \delta_{z_k}$  (how the reference template is computed is described in section 3.4). The distance between  $\mu$  and  $\nu$  given by (3) in a discrete setting becomes:

$$d_{LOT,\sigma}(\mu, \nu)^2 = \min_{f \in \Pi_{OT}(\sigma, \mu), g \in \Pi_{OT}(\sigma, \nu)} \min_h \left\{ \sum_{i=1}^{N_\mu} \sum_{j=1}^{N_\nu} \sum_{k=1}^{N_\sigma} h_{k,i,j} |x_i - y_j|^2 : \text{where } h_{k,i,j} \geq 0, \sum_{i=1}^{N_\mu} h_{k,i,j} = g_{k,j} \text{ and } \sum_{j=1}^{N_\nu} h_{k,i,j} = f_{k,i} \right\} \quad (6)$$

We remark that the sets of optimal transportation plans,  $\Pi_{OT}(\sigma, \mu)$  and  $\Pi_{OT}(\sigma, \nu)$ , typically have only one element, so there is only one possibility for  $f$ , and  $g$ . See Figure 4 for a visualization. Usually there is more than one possibility for  $h$ , in the case of particle measures.

We now introduce the ‘distance’ which is an approximation of the one above (hence we denote it as  $d_{aLOT}$ ) and which is used in most computations in this paper. Namely, given OT plans  $f$  and  $g$ , as indicated in Figure 5, we replace the  $f$ -image of the particle at  $z_k$ , namely the measure  $\sum_i f_{k,i} \delta_{x_i}$  by one particle at the center of mass, namely the measure  $q_k \delta_{\bar{x}_k}$ . We recall from Section 2.2 that if  $\sigma, \mu$  and  $\nu$  were measures which had a density function (and hence no particles) then there exists an optimal transportation map and the image of any  $z$  is just a single  $x$ . On the discrete level we have an approximation of that situation and typically the  $f$ -image of the particle at  $z_k$  is spread over a few nearby particles. Thus the error made by replacing the  $f$ -image by a single particle at the images center of mass is typically small.

To precisely define the new distance, let  $f$  be an optimal transportation plan between

$\sigma = \sum_{k=1}^{N_\sigma} q_k \delta_{z_k}$  and  $\mu = \sum_{i=1}^{N_\mu} m_i \delta_{x_i}$  obtained in (5), and  $g$  is an optimal transportation plan between  $\sigma = \sum_{k=1}^{N_\sigma} q_k \delta_{z_k}$  and  $\nu = \sum_{j=1}^{N_\nu} p_j \delta_{y_j}$ . Then

$$\bar{x}_k = \frac{1}{q_k} \sum_{i=1}^{N_\mu} f_{k,i} x_i \text{ and } \bar{y}_k = \frac{1}{q_k} \sum_{j=1}^{N_\nu} g_{k,j} y_j \quad (7)$$

are the centroids of the forward image of the particle  $q_k \delta_{z_k}$  by the transportation plans  $f$  and  $g$  respectively (see Figure 5). We define

$$d_{aLOT,\sigma}(\mu, \nu)^2 = \min_{\substack{f \in \Pi_{OT}(\sigma, \mu) \\ g \in \Pi_{OT}(\sigma, \nu)}} \sum_{k=1}^{N_\sigma} q_k |\bar{x}_k - \bar{y}_k|^2 \quad (8)$$

We clarify that computing (8) in practice does not require a minimization over  $f$  and  $g$ , since  $f$  and  $g$  are unique with probability one. The reason for that is that in the discrete case the condition for optimality of transportation plans can be formulated in terms of cyclic monotonicity (see [40]). The nonuniqueness of the OT plan can only happen if the inequalities in some cyclic monotonicity conditions become equalities, which means that the coordinates of particles satisfy an algebraic equation. And this can only happen with probability zero. For example, if both measures have exactly two particles of the same mass then the condition for nonuniqueness is that  $|x_1 - y_1|^2 + |x_2 - y_2|^2 = |x_2 - y_1|^2 + |x_1 - y_2|^2$ .

Thus we compute

$$d_{aLOT,\sigma}(\mu, \nu)^2 = \sum_{k=1}^{N_\sigma} q_k |\bar{x}_k - \bar{y}_k|^2. \quad (9)$$

One should notice that the  $d_{aLOT,\sigma}$  distance between two particle measures may be zero. This is related to the ‘resolution’ achieved by the base measure  $\sigma$ . In particular if  $\sigma$  has more particles then it is less likely that  $d_{aLOT,\sigma}(\mu, \nu) = 0$  for  $\mu \neq \nu$ . It is worth observing that  $d_{aLOT,\sigma}(\mu, \nu)^2$  computed in (9) is always less than or equal to  $d_{LOT,\sigma}(\mu, \nu)^2$  introduced in equation (6).

Furthermore if we consider measures with continuous density  $\mu, \nu$  and approximate them by particle measures  $\mu_n, \nu_n$  then as the number of particles goes to infinity  $d_{LOT,\sigma_n}(\mu_n, \nu_n) \rightarrow d_{LOT,\sigma}(\mu, \nu)$  where the latter object is defined in (2). This follows from the stability of optimal transport [40].

Let us also remark that, while we use the linear programming problem as described in (5) to compute the optimal transportation plan, other methods can be used as well. In particular the LOT distance is applicable even when a different approach to computing the OT plan is used.

### 3.4 Template selection

Selecting the template (reference)  $\sigma$  above is important since, in our experience, a template that is distant from all images (particle approximations) is likely to increase the difference between  $d_W$  and  $d_{aLOT}$ . In the results presented below, we use a reference template computed from the ‘average’ image. To compute such average image, we first align all images to remove translation, rotation, and horizontal and vertical flips, as described in Section 3.3. All images are then averaged, and the particle approximation algorithm described above is used to obtain a particle representation of the average image. We denote this template as  $\sigma$  and note that it will contain  $N_\sigma$  particles. Once the particle approximation for the template is computed, it is also normalized to a standard position and orientation as described above.

### 3.5 Isometric linear embedding

When applying our approach to a given a set of images (measures)  $I_1, \dots, I_M$  we first compute the particle approximation for each image with the algorithm described in section 3.1. We then compute a template  $\sigma$  as described in section 3.4 and compute the OT distance

(5) between a template  $\sigma$  (also chosen to be a particle measure) and the particle approximation of each image. Once these are computed, the LOT distance between any two particle sets is given by (9).

The lower bound of the linear optimal transportation distance  $d_{aLOT,\sigma}$  defined in equation (9) provides a method to map a sample measure  $\nu_n$  (estimated from image  $I_n$ ) into a linear space. Let  $\nu_n = \sum_{j=1}^{N_{\nu_n}} m_j \delta_{y_j}$  and recall that the reference measure is  $\sigma = \sum_{k=1}^{N_\sigma} q_k \delta_{z_k}$ . The linear embedding is obtained by applying the discrete transportation map between the reference measure  $\sigma$  and  $\nu_n$  to the coordinates  $y_j$  via

$$\mathbf{x}_n = \left( \sqrt{q_1} a_n^1 \cdots \sqrt{q_{N_\sigma}} a_n^{N_\sigma} \right)^T \quad (10)$$

where  $a_n^k$  is the centroid of the forward image of the particle  $q_k \delta_{z_k}$  by the optimal transportation plan,  $g_{k,j}$ , between images  $\sigma$  and  $\nu_n$ :

$$a_n^k = \sum_{j=1}^{N_{\nu_n}} g_{k,j} y_j / q_k \quad (11)$$

This results in an  $N_\sigma$ -tuple of points in  $\mathbb{R}^2$  which we call the linear embedding  $\mathbf{x}_n$  of  $\nu_n$ . That is,  $\mathbf{x}_n \in \mathbb{R}^{N_\sigma \times 2}$ . We note that the embedding is interpretable in the sense that any point in this space can be visualized by simply plotting the vector coordinates (each in  $\mathbb{R}^2$ ) in the image space  $\Omega$ .

## 4 Statistical analysis

When a linear embedding for the data can be assumed, standard geometric data processing techniques such as principal component analysis (PCA) can be used to extract and visualize major trends of variation in morphology [8,33,38]. Briefly, given a set of data points  $\mathbf{x}_n$ , for  $n = 1, \dots, M$ , we can compute the covariance matrix  $S = \frac{1}{M} \sum_n (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$ , with  $\bar{\mathbf{x}} = \frac{1}{M} \sum_{n=1}^M \mathbf{x}_n$  representing center of the entire data set. PCA is a method for computing the major trends (directions over which the projection of the data has largest variance) of a dataset via the solution of the following optimization problem:

$$\mathbf{w}_{PCA}^* = \arg \max_{\|\mathbf{w}\|=1} \mathbf{w}^T S \mathbf{w} \quad (12)$$

The problem above can be solved via eigenvalue and eigenvector decomposition, with each eigenvector corresponding to a major mode of variation in the dataset (it's corresponding eigenvalue is the variance of the data projected over that direction).

In addition to visualizing the main modes of variation for a given dataset, important applications involve visualizing the modes of variation that best discriminate between two or more separate populations (e.g. as in control vs. effect studies). To that end, we apply the methodology we developed in [41], based on the well known Fisher linear discriminant analysis (FLDA) technique [14]. As explained in [41], simply applying the FLDA technique in morphometry visualization problems can lead to erroneous interpretations, since the directions computed by the FLDA technique are not constrained to pass through the data. To

alleviate this effect we modified the method as follows. Briefly, given a set of data points  $\mathbf{x}_n$ , for  $n = 1, \dots, N$ , with each index  $n$  belonging to class  $c$ , we modified the original FLDA by adding a least squares-type representation penalty in the function to be optimized. The representation constrained optimization can then be reduced to

$$\mathbf{w}_{LDA}^* = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T S_T \mathbf{w}}{\mathbf{w}^T (S_W + \alpha \mathbf{I}) \mathbf{w}} \quad (13)$$

where  $S_T = \sum_n (\mathbf{x}_n - \hat{\mathbf{x}})(\mathbf{x}_n - \hat{\mathbf{x}})^T$  represents the ‘total scatter matrix’,  $S_W = \sum_c \sum_{n \in c} (\mathbf{x}_n - \hat{\mathbf{x}}_c)(\mathbf{x}_n - \hat{\mathbf{x}}_c)^T$  represents the ‘within class scatter matrix’,  $\hat{\mathbf{x}}_c$  is the center of class  $c$ . The solution for the problem above is given by the well-known generalized eigenvalue decomposition  $S_T \mathbf{w} = \lambda (S_W + \alpha \mathbf{I}) \mathbf{w}$  [7]. In short, the penalized LDA method above seeks to find the direction  $\mathbf{w}$  that has both a low reconstruction error, but that best discriminates between two populations. We use the method described in [41] to select the penalty weight  $\alpha$ .

## 5 Computational results

In this section, we describe results of applying the LOT method to quantify the statistical variation of three types of datasets. We analyze (sub) cellular protein patterns and organelles from microscopy images, visualize the variation of shape and brightness in a galaxy image database and characterize the variation of expressions in a facial image database. We begin by introducing the datasets used. For the cellular image datasets, we then show results that 1) evaluate the particle approximation algorithm described earlier, 2) evaluate how well the LOT distance approximates the OT distance, 3) evaluate the discrimination power of the LOT distance (in comparison to more traditional feature-based methods), and 4) use the linear embedding provided by the LOT to visualize the geometry (summarizing trends), including discrimination properties. For both the galaxy and facial images, we evaluate the performance of the particle approximation and discrimination power of the LOT metric, and visualize the discriminating information. For the facial image dataset, we also visualize the summarizing trends of variation of facial expressions. In the case of facial expressions, the results show that LOT obtains similar quantification of expression as the original paper [36], yet LOT has the distinct advantage of being landmark free.

### 5.1 Datasets and pre-processing

The biological imaging dataset has two sub-datasets. The nuclear chromatin patterns were extracted from histopathology images obtained from tissue sections. Tissue for imaging was obtained from the tissue archives of the Children’s Hospital of Pittsburgh of the University of Pittsburgh Medical Center. The extraction process included a semi-automatic level set-based segmentation, together with a color to gray scale conversion and pixel normalization. The complete details related to this dataset are available in our previous work [43], [42]. The dataset we use in this paper consists of five human cases of liver hepatoblastoma (HB), a liver tumor in children, each containing adjacent normal liver tissue (NL). In total, 100 nuclei were extracted from each case (50 NL, 50 HB). The second sub dataset we use are fluorescent images of 2 golgi proteins: giantin and GPP130. The dataset is described in detail in [9]. In total, we utilize 81 GPP and 66 giantin protein patterns. The galaxy dataset we use contain two types of galaxy images including 225 Elliptical galaxies and 223 Spiral galaxies. The dataset is described in [34]. The facial expression dataset is the same as described in [36]. We use only the normal and smiling expressions, with each expression containing 40 images. Sample images for the nuclear chromatin, golgi, galaxy and facial expression datasets are show in Figure 6(A), (B), (C) and (D) respectively.

## 5.2 Particle approximation error

Here we quantify how well our particle-based algorithm described above can approximate the different datasets we use in this paper. Instead of an absolute figure, of interest is the OT error between each image and its particle approximation, relative to the OT distances that are present in a dataset. Figure 7(A) can give us some indication of the relative (in percentage) error for the particle approximation of each image in each dataset. For each

image in a dataset, we compute the error  $e_i = \frac{\phi_i}{\frac{1}{M} \sum_{j=1, \dots, M} d_{i,j}^{OT}}$  where  $d_{i,j}$  is the OT distance computed through equation (5), and  $\phi_i$  is the upper bound on the OT distance between the particle approximation of image  $i$  and the image  $i$  (defined in equation (16) in the appendix). The computation is repeated for  $M$  images chosen randomly from a given dataset. Figure 7(A) contains the histogram of the errors for all the datasets. The x-axis represents the relative error in percentage, and the y-axis represents the portion of samples that has the corresponding relative error. Though, by definition, we can always reduce the approximation errors by adding more particles, considering the computational cost, we use 500 initial particles for the nuclei, golgi and galaxy data sets, and 2000 initial particles for the face data set. The data shows that the average errors were 14.1% for the nuclear chromatin dataset, 3.2% for the golgi protein dataset, 13.1% for the galaxy image dataset and 18.5% for the face data set.

## 5.3 Difference between OT and LOT

We compare the LOT distance given in equation (9) with the OT distance, defined in equation (5). To that end, we compute the full OT distance matrix (all pairwise distances)

for both biological image datasets and compare it to the LOT distance via:  $e_{OT,LOT} = \frac{d_{OT} - d_{LOT}}{d_{OT}}$ . Figure 7(B) contains the relative error for the nuclear chromatin and golgi. The average absolute error for the nuclear chromatin data is  $e_{OT,LOT} = 2.94\%$ , and  $e_{OT,LOT} = 3.28\%$  for the golgi dataset.

## 5.4 Discrimination power

In [43], we have shown that the OT metric can be used to capture the nuclear morphological information to classify different classes of liver and thyroid cancers. We now show that the linear OT approximation we propose above can be used for discrimination purposes with little or no loss in accuracy when compared to traditional feature-based approaches. To test classification accuracy we use a standard implementation of the support vector machine method [7] with a radial basis function (RBF) kernel.

For the nuclear chromatin in the liver cancer dataset we followed the same leave one case out cross validation approach described in [43]. The classification results are listed in Table 1. For comparison, we also computed classification results using a numerical feature approach where the same 125 numerical features (including shape parameters, Haralick features, and multi-resolution-type features) as described in [43], were used. Stepwise discriminant analysis was used to select the significant features for classification (12 of them were consequently selected). In addition, we also provide classification accuracies utilizing the method described in [28], which we denote here as EMD-L1. The EMD-L1 in this case was applied to measure the  $L_1$  weighted earth mover's distance (EMD) between image pairs. The radius above which pixels displacements were not computed was set to 15 for fast computation. When using the EMD-L1 method, each image was downsampled by four, after Gaussian filtering, to allow for the computations to be performed in a reasonable time frame (see discussion section for a comparison and discussion of computation times).

Classification results are shown in Table 1. We note that because we used a kernel-based SVM method, the only difference between all implementations were the pairwise distances

computed (OT vs. LOT vs. EMD-L1 vs. feature-based). In Table 1, each row corresponds to a test case, and the numbers correspond to the average classification accuracy (per nucleus) for normal liver and liver cancer. The first column contains the classification accuracy for the feature-based approach, the second column the accuracy for the OT metric, the third column the accuracy for the LOT metric, and the final column contains the computations utilizing the EMD-L1 approach [28]. We followed a similar approach for comparing classification accuracies in the golgi protein dataset. In this case we utilized the feature set described in [9]. This feature set was specifically designed for classification tasks of this type. A five fold cross validation strategy was used. Results are shown in Tables 2 and 3.

We also computed the LOT metric for the galaxy and facial expression datasets and applied the same support vector machine (SVM) strategy. In the galaxy case we utilized the feature set described in [34]. A five fold cross validation accuracy was used. Compared with the accuracy of the feature-based approach (93.6%), the classification accuracy of LOT metric was 87.7%. For the facial expression data set, the classification was performed based on SVM with the LOT metric, and a 90% classification accuracy was obtained.

### 5.5 Visualizing summarizing trends and discriminating information

We applied the PCA technique as described in section 4 to the nuclei and facial expression data sets. The first three modes of variation for the nuclear chromatin datasets (containing both normal and cancerous cells) are displayed in Figure 8(A). The first three modes of variation for the facial expression dataset are shown in Figure 8(B). For both datasets, the first three modes of variation correspond to over 99% of the variance of the data. The modes of variation are displayed from the mean to plus and minus four times times the standard deviation of each corresponding mode. For the chromatin dataset we can visually detect size (first mode), elongation (second mode), and differences in chromatin concentration, from the center to the periphery of the nucleus (third mode). In the face dataset, since we did not normalize for size of the face, one can detect variations in size (first mode), facial hair texture (second mode), as well as face shape (third mode). The variations detected from the second and third modes are similar to the results reported in [36].

We also applied the method described in Section 4 to visualize the most significant differences between the two classes in each dataset. The most discriminating modes for the nuclear chromatin dataset, golgi dataset, galaxy and the facial expression dataset are shown in Figure 9(A,B,C,D), respectively. In the nuclear chromatin case, the normal tissue cells tend to look more like the images on the left, while the cancerous ones tend to look more like the images on the right. In this case we can see that the most discriminating effect seems to be related to differences in chromatin placement (near periphery vs. concentrated at the center). The  $p$  value for this direction was 0.019. For the golgi protein dataset, the discriminating direction ( $p = 0.049$ ) shows that the giantin golgi protein tends to be more scattered than the gpp protein, which tends to be more elongated in its spatial distribution. For the galaxy dataset, the discriminant direction ( $p = 0.021$ ) seems to vary from a spiral structure to a bright dense disk. For the facial expression dataset, the discriminating direction ( $p = 0.011$ ) shows a trend from a smiling face to a serious or neutral facial expression.

## 6 Summary and Discussion

We described a new method for quantifying and visualizing morphological differences in a set of images. Our approach called the LOT is based on the optimal transportation framework with quadratic cost, and we utilize a linearized approximation based on a tangent space representation to make the method amenable to large datasets. Our current implementation is based on a discrete ‘particle’ approximation of each pattern, with

subsequent linear programming optimization, and is therefore best suited for images which are sparse. We evaluated the efficacy of our methods in several aspects. As our results show, the error between an image and its particle approximation, relative to the distances in the dataset, are relatively small for a moderate number of particles when the images are sparse (e.g. golgi protein images), and can be made small for general images if more time is allowed for computation. We also showed that the LOT distance closely approximates the standard OT distance between particle measures, with absolute percentage errors on the order of a few percent in the datasets we used. In experiments not shown, we also evaluated the reproducibility of our particle-based LOT computation with respect to the random particle initializations required by algorithm. These experiments revealed that the average coefficient of variation was on the order of a couple of percent. Finally, we also evaluated how well the LOT distance described can perform in classifying sets of images in comparison to standard feature-based methods, the traditional OT distance [43], and the EMD-L1 method of [28]. Overall, considering all classification tests, no significant loss of accuracy was detected.

A major advantage of the LOT framework is the reduced cost of computing the LOT distance between all image pairs in a dataset. For a database of  $M$  images, section 2.3 explains that the number of transportation related optimization problems that need to be solved for computing all pairwise distances is  $M$ . In comparison, to our knowledge, the number of transportation related optimizations necessary for this purpose of all other available methods is  $M(M-1)/2$ . In concrete terms, if all computations were to be performed using a single Apple laptop computer with 2GB or RAM and a dual processor of 2.2GHz, the total computation time for producing all pairwise distances for computing the results shown in Table 1 of our paper, for example, would be approximately 4.1 hours for the LOT method, 15.6 hours for the EMD-L1 method [28], and 1200 hours for the OT method described in [43]. We clarify that in order to utilize the EMD-L1 method images had to be downsampled to allow the computations to be performed in reasonable times. The time quoted above (and results shown in Tables 1, and 3) was computed by downsampling each image by a factor of four (after spatial filtering). We clarify again, however, that we have used a generic linear programming package for solving the OT optimizations associated with our LOT framework. The computation times for the LOT framework could be decreased further by utilizing linear programming techniques more suitable for this problem.

In addition to fast computation, a significant innovation of the LOT approach is the convenient isometric linear embedding it provides. In contrast to other methods that also obtain linear embeddings, see for example [31,37], the embedding LOT space allows for one to synthesize and visualize any point in the linear space as an image. Therefore the LOT embedding can facilitate visualization of the data distribution (via PCA) as well as differences between distributions (via penalized LDA). Hence, it allows for the automatic visualization of differences in both shape and texture in different classes of images. In biological applications, such meaningful visualizations can be important in helping elucidate effects and mechanisms automatically from the image data. For example, the results with the nuclear chromatin dataset suggest that, in this disease, cancer cells tend to have their chromatin more evenly spread (euchromatin) throughout the nucleus, whereas normal cells tend to have their chromatin in more compact form (heterochromatin). This suggests that the loss of heterochromatin is associated with the cancer phenotype in this disease. This finding is corroborated by earlier studies [42,43] (which used different methodology), as well as other literature which suggests that cancer progression is associated with loss of heterochromatin [25, 13].

We note that the use of the quadratic cost in (1) allows for a Riemannian geometric interpretation for the space of images. With this point of view, the LOT framework we

describe can be viewed as a tangent plane approximation of the OT manifold. Thus the linear embedding we mentioned above can be viewed as a projection onto the tangent plane. While computing theoretical bounds for the difference between the OT and LOT distances is difficult since it would depend on the curvature of the OT manifold (which we have no way of estimating for a real distribution of images), several important facts about this approximation are worth noting. Firstly, we note that the LOT is a proper mathematical distance on its own, even when the tangent plane approximation is poor. Secondly, the projection is a one to one mapping. Thus information cannot be lost by projecting two images onto the same point. Finally, we note that utilizing the LOT distances (instead of the OT ones) results in no significant loss in classification accuracy. This suggests, that for these datasets, no evidence is available to prefer the OT space versus the LOT one.

As mentioned above, the LOT embedding obtained via the tangent space approximation can facilitate the application of geometric data analysis methods such as PCA to enable one to visualize interesting variations in texture and shape in the dataset. As compared to applying the PCA method directly on pixel intensities (data not shown for brevity), the PCA method applied with the use of the LOT method yields sharper and easier to interpret variations, since it can account for both texture and shape variations via the transportation of mass. As already noted above, several other graph-based methods also exist for the same purpose [31,37]. However, in contrast to our LOT method, these are not analytical meaning that only measured images can be visualized. Moreover, the LOT distances we provide here can be used as ‘local’ distances, based on which such graphs can be constructed. Thus both classes of methods could be used jointly, for even better performance (one example is provided in [30]).

The LOT framework presented here depends on a ‘particle’ approximation of the input images. The advantage of such a particle approximation is that the underlying OT optimization becomes a linear program whose global optimum can be solved exactly with standard methods. Drawbacks include the fact that such approach can become computationally intensive for non sparse images. We investigated the average classification accuracy for the experiments encountered in the previous section as a function of the number of particles used to approximate each image. For the experiment involving nuclear chromatin patterns, for example, the average classification accuracy for the LOT method using 300, 500, and 900 particles was 79%, 83.3%, and 83.4%, respectively. We note that for the golgi protein dataset, utilizing the LOT with 100 particles produced nearly the same accuracy as the ones reported in Table 2. These results, together with the already provided comparison to other methods, suggest the particle approximation was sufficient to classify these datasets.

Finally, we wish to point out several limitations inherent in the approach we describe here. First, as already noted, the particle approximation is best able to handle images which are sparse. For images that are not sparse, many more particles are needed to approximate the image well, thus increasing significantly the computational cost of the method. In addition, like all transportation related distances, our approach is not able to handle transport problems which include boundaries. Instead the approach is best suited for analyzing images whose intensities can be viewed as a distribution of mass that is ‘free’ to move about the image. In addition, we note that transportation-type distances are not good at telling the difference between smooth versus punctate patterns. For this specific task, other approaches (such as Fourier analysis, for example), may be more fruitful. Finally, we note that for some types of images, the traditional OT and the LOT can yield very different distances. This could occur when the images to be compared have very sharp transitions, of high frequency, and if the phase of these transitions are mismatched. We note that this is rarely the case in



the images we analyze in this paper. These and other aspects of our current LOT framework will be the subject of future work.

## Acknowledgments

The authors wish to thank the anonymous reviewers for helping significantly improve this paper. WW, SB, and GKR acknowledge support from NIH grants GM088816 and GM090033 (PI GKR) for supporting portions of this work. DS was also supported by NIH grant GM088816, as well as NSF grant DMS-0908415. He is also grateful to the Center for Nonlinear Analysis (NSF grant DMS-0635983 and NSF PIRE grant OISE-0967140) for its support.

## References

1. Ambrosio, L.; Gigli, N.; Savaré, G. Lectures in Mathematics ETH Zürich. 2. Birkhäuser Verlag; Basel: 2008. Gradient flows in metric spaces and in the space of probability measures.
2. Angenent S, Haker S, Tannenbaum A. Minimizing flows for the Monge-Kantorovich problem. *SIAM J Math Anal.* 2003; 35(1):61–97. (electronic).
3. Barrett JW, Prigozhin L. Partial  $L^1$  Monge-Kantorovich problem: variational formulation and numerical approximation. *Interfaces Free Bound.* 2009; 11(2):201–238.
4. Beg M, Miller M, Trounev A, Younes L. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision.* 2005; 61(2):139–157.
5. Benamou JD, Brenier Y. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numer Math.* 2000; 84(3):375–393.
6. Bengtsson E. Fifty years of attempts to automate screening for cervical cancer. *Med Imaging Tech.* 1999; 17:203–210.
7. Bishop, CM. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer; 2006.
8. Blum H, et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form.* 1967; 19(5):362–380.
9. Boland MV, Murphy RF. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela cells. *Bioinformatics.* 2001; 17(12):1213–23. [PubMed: 11751230]
10. do Carmo, MP. *Mathematics: Theory & Applications.* Birkhäuser Boston Inc; Boston, MA: 1992. Riemannian geometry. Translated from the second Portuguese edition by Francis Flaherty
11. Chef'd'hotel, C.; Bousquet, G. Intensity-based image registration using earth mover's distance. *Proceedings of SPIE;* 2007. p. 65122B
12. Delzanno GL, Finn JM. Generalized Monge-Kantorovich optimization for grid generation and adaptation in  $L_p$ . *SIAM J Sci Comput.* 2010; 32(6):3524–3547.
13. Dialynas GK, Vitalini MW, Wallrath LL. Linking heterochromatin protein 1 (hp1) to cancer progression. *Mutat Res.* 2008; 647(1–2):13–20. [PubMed: 18926834]
14. Fisher RA. The use of multiple measurements in taxonomic problems. *Annals of Eugenics.* 1936; 7:179–188.
15. Gardner M, Sprague B, Pearson C, Cosgrove B, Bicek A, Bloom K, Salmon E, Odde D. Model convolution: A computational approach to digital image interpretation. *Cellular and molecular bioengineering.* 2010; 3(2):163–170. [PubMed: 20461132]
16. Grauman, K.; Darrell, T. Fast contour matching using approximate earth mover's distance. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR;* 2004; 2004.
17. Haber E, Rehman T, Tannenbaum A. An efficient numerical method for the solution of the  $L_2$  optimal mass transfer problem. *SIAM J Sci Comput.* 2010; 32(1):197–211. [PubMed: 21278828]
18. Haker S, Zhu L, Tannenbaum A, Angenent S. Optimal mass transport for registration and warping. *Intern J Comp Vis.* 2004; 60(3):225–240.
19. Kong J, Sertel O, Shimada H, Boyer KL, Saltz JH, Gurcan MN. Computer-aided evaluation of neuroblastoma on whole slide histology images: classifying grade of neuroblastic differentiation. *Pattern Recognition.* 2009; 42:1080–1092.

20. Ling H, Okada K. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2007;840–853. [PubMed: 17356203]
21. Lloyd SP. Least squares quantization in pcm. *IEEE Trans Inf Theory*. 1982; 28(2):129–137.
22. Loo L, Wu L, Altschuler S. Image-based multivariate profiling of drug responses from single cells. *Nature Methods*. 2007; 4(5):445–454. [PubMed: 17401369]
23. Methora S. On the implementation of a primal-dual interior point method. *SIAM Journal on Scientific and Statistical Computing*. 1992; 2:575–601.
24. Miller MI, Priebe CE, Qiu A, Fischl B, Kolasny A, Brown T, Park Y, Ratnanather JT, Busa E, Jovicich J, Yu P, Dickerson BC, Buckner RL. Collaborative computational anatomy: an mri morphometry study of the human brain via diffeomorphic metric mapping. *Hum Brain Mapp*. 2009; 30(7):2132–41. [PubMed: 18781592]
25. Moss TJ, Wallrath LL. Connections between epigenetic gene silencing and human disease. *Mutat Res*. 2007; 618(1–2):163–74. [PubMed: 17306846]
26. Orlin JB. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*. 1993; 41(2):338–350.
27. Pele, O.; Werman, M. A linear time histogram metric for improved sift matching. *ECCV*; 2008.
28. Pele, O.; Werman, M. Fast and robust earth mover's distances. *Computer Vision, 2009 IEEE 12th International Conference on; IEEE*. 2009. p. 460-467.
29. Pincus Z, Theriot JA. Comparison of quantitative methods for cell-shape analysis. *J Microsc*. 2007; 227(Pt 2):140–156. [PubMed: 17845709]
30. Rohde GK, Ribeiro AJS, Dahl KN, Murphy RF. Deformation-based nuclear morphometry: capturing nuclear shape variation in hela cells. *Cytometry A*. 2008; 73(4):341–50. [PubMed: 18163487]
31. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*. 2000; 290(5500):2323–2326.10.1126/science.290.5500.2323 [PubMed: 11125150]
32. Rubner Y, Tomassi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. *Intern J Comp Vis*. 2000; 40(2):99–121.
33. Rueckert D, Frangi AF, Schnabel JA. Automatic construction of 3-d statistical deformation models of the brain using nonrigid registration. *IEEE Trans Med Imag*. 2003; 22(8):1014–1025.
34. Shamir, L. Monthly Notices of the Royal Astronomical Society. 2009. Automatic morphological classification of galaxy images.
35. Shirdhonkar, S.; Jacobs, D. Approximate earth mover's distance in linear time. *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*; 2008; 2008.
36. Stegmann M, Ersboll B, Larsen R. Fame-a flexible appearance modeling environment. *IEEE Transactions on Medical Imaging*. 2003; 22(10):1319–1331. [PubMed: 14552585]
37. Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science*. 2000; 290(5500):2319–2323.10.1126/science.290.5500.2319 [PubMed: 11125149]
38. Vaillant M, Miller M, Younes L, Trouvé A. Statistics on diffeomorphisms via tangent space representations. *NeuroImage*. 2004; 23:S161–S169. [PubMed: 15501085]
39. Villani, C. *Graduate Studies in Mathematics*. Vol. 58. American Mathematical Society; Providence, RI: 2003. Topics in optimal transportation.
40. Villani, C. *Optimal transport, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Vol. 338. Springer-Verlag; Berlin: 2009. URL <http://dx.doi.org/10.1007/978-3-540-71050-9>. Old and new
41. Wang W, Mo Y, Ozolek JA, Rohde GK. Penalized fisher discriminant analysis and its application to image-based morphometry. *Pattern Recognition Letters*. (accepted, 2011).
42. Wang W, Ozolek J, Rohde G. Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images. *Cytometry Part A*. 2010; 77(5):485–494.

43. Wang W, Ozolek J, Slepcev D, Lee A, Chen C, Rohde G. An optimal transportation approach for nuclear structure-based pathology. *IEEE Transactions on Medical Imaging*. 2011; 30(3):621–631. [PubMed: 20977984]
44. Yang L, Chen W, Meer P, Salaru G, Goodell L, Berstis V, Foran D. Virtual microscopy and grid-enabled decision support for large scale analysis of imaged pathology specimens. *IEEE Trans Inf Technol Biomed*. 2009

## Appendix: Particle approximation algorithm

Here we present the details of the particle approximation algorithm outlined in section 3.1. The goal of the algorithm is to approximate the given probability measure,  $\mu$  on  $\Omega$  by a particle measure with approximately  $N$  particles, where  $N$  is given. For images where the estimated error of the initial approximation is much larger than the typical error over the given dataset the number of particles is increased to reduce the error, while for images where the error is much smaller than typical the number of particles is reduced in order to save time when computing the OT distance. The precise criterion and implementation of adjusting the number of particles is described in step 8 below.

In our application the measure  $\mu$  represents the image. It too can be thought as a particle

measure particle measure itself  $\mu = \sum_{j=1}^{N_\mu} m_j \delta_{y_j}$  where  $N_\mu$  is the number of pixels,  $y_j$  the coordinates of the pixel centers, and  $m_j$  the intensity values. Our goal however is to approximate it with a particle measure with a lot fewer particles. Below we use the fact that

the *restriction* of the measure  $\mu$  to a set  $V$  is defined as  $\mu|_V = \sum_{i=1, x_i \in V}^{N_\mu} m_i \delta_{x_i}$ . The backbone of the algorithm rests on the following mathematical observations.

### Observation 1. The best approximation for fixed particle locations

Assume that  $x_1, \dots, x_N$  are fixed. Consider the Voronoi diagram with centers  $x_1, \dots, x_N$ . Let  $V_j$  be the cell of the Voronoi diagram that correspond to the center  $x_j$ . Then among all

probability measures of the form  $\mu = \sum_{i=1}^N m_i \delta_{x_i}$ , the one that approximates  $\mu$  the best is

$$\sum_{i=1}^N \mu(V_i) \delta_{x_i} = \operatorname{argmin} \left\{ d_w(\mu, \mu_N) : \mu_N = \sum_{i=1}^N m_i \delta_{x_i} \text{ with } m_i \geq 0 \text{ and } \sum_{i=1}^N m_i = \mu(\Omega) \right\} \quad (14)$$

Let us remark that if  $\mu$  is a particle measure as above then  $\mu(V_j)$  is just the sum of intensities of all pixels whose centers lie in  $V_j$ .

Given a measure  $\mu$  supported on  $\Omega$  we define the center of mass to be

$$x(\mu) = \frac{1}{\mu(\Omega)} \int_{\Omega} x d\mu. \quad (15)$$

### Observation 2. The best local ‘recentering’ of particles

It is easy to prove that among all one-particle measures the one that approximates  $\mu$  the best is the one set at the center of mass of  $\mu$ :

$$\mu(\Omega)\delta_{x(\mu)} = \operatorname{argmin}\{d_w(\mu, \mu(\Omega)\delta_y); y \in \mathbb{R}^n\}.$$

We can apply this to ‘recenter’ the delta measures in each Voronoi cell:

$$\sum_{i=1}^N \mu(V_i) \delta_{x(V_i)} = \operatorname{argmin} \left\{ \sum_{i=1}^k d_w(\mu|_{V_i}, \mu(V_i)\delta_{y_i}); y_i \in \mathbb{R}^n \right\}$$

Here  $\mu|_{V_i}$  is the restriction of measure  $\mu$  to the set  $V_i$ . More precisely, for  $A \subset \Omega$ ,  $\mu|_{V_i}(A) = \mu(A \cap V_i)$ .

**Observation 3. The error for the current approximation is easy to estimate**

Given a particle approximation as above one can compute a good upper bound on the error. In particular

$$\begin{aligned} d_w^2 \left( \mu, \sum_{i=1}^N \mu(V_i) \delta_{x(V_i)} \right) &\leq \sum_{i=1}^N d_w^2 \left( \mu|_{V_i}, \mu(V_i) \delta_{x(V_i)} \right) \\ &= \sum_{i=1}^N \int |x - x(\mu|_{V_i})|^2 d\mu|_{V_i}. \end{aligned} \tag{16}$$

If  $\mu = \sum_{j=1}^{N_\mu} m_j \delta_{y_j}$ , the upper bound on the right-hand side becomes  $\sum_{i=1}^N \sum_{j=1, y_j \in V_i}^{N_\mu} m_j |y_j - \bar{x}_i|^2$

where  $\bar{x}_i = \left( \sum_{j=1, y_j \in V_i}^{N_\mu} m_j y_j \right) / \sum_{j=1, y_j \in V_i}^{N_\mu} m_j$ . We should also note that the estimate on the right-hand side gives us very useful information on ‘local’ error in each cell  $V_i$ . This enables us to determine which cells need to be refined when needed.

Based on these observations, we use the following ‘weighted Lloyd’ algorithm to produce a particle approximation to measure  $\mu$ . The idea is to use enough particles to approximate each image well (according to the transport criterion), but not more than necessary given a chosen accuracy. The steps of our algorithm are:

1. Distribute  $N$  particles (with  $N$  an arbitrarily chosen number),  $x_1, \dots, x_N$ , over the domain  $\Omega$  by weighted random sampling (without replacement) the measure  $\mu$  with respect to the intensity values of the image.
2. Compute the Voronoi diagram for centers  $x_1, \dots, x_N$ . Based on Observation 1, set  $m_i = \mu(V_i)$ . The measure  $\mu_N = \sum_{i=1}^N \mu(V_i) \delta_{x_i}$  is the current approximation.
3. Using Observation 2, we recenter the cells by setting  $x_{i,new} = x(\mu|_{V_i})$  (the center of mass of  $\mu$  restricted to cell  $V_i$ ).
4. Repeat steps 2. and 3. until the algorithm stabilizes. (the change of error upper bound is less than 0.5% in the sequential steps).
5. Repeat the steps 1 to 4 10 times for each image and choose the approximation with lowest error.
6. We then seek to improve the approximation in regions in the image which have not been well approximated by the step above. We do so by introducing more particles in the Voronoi cells  $V_i$  where the cost of transporting the intensities in that cell to

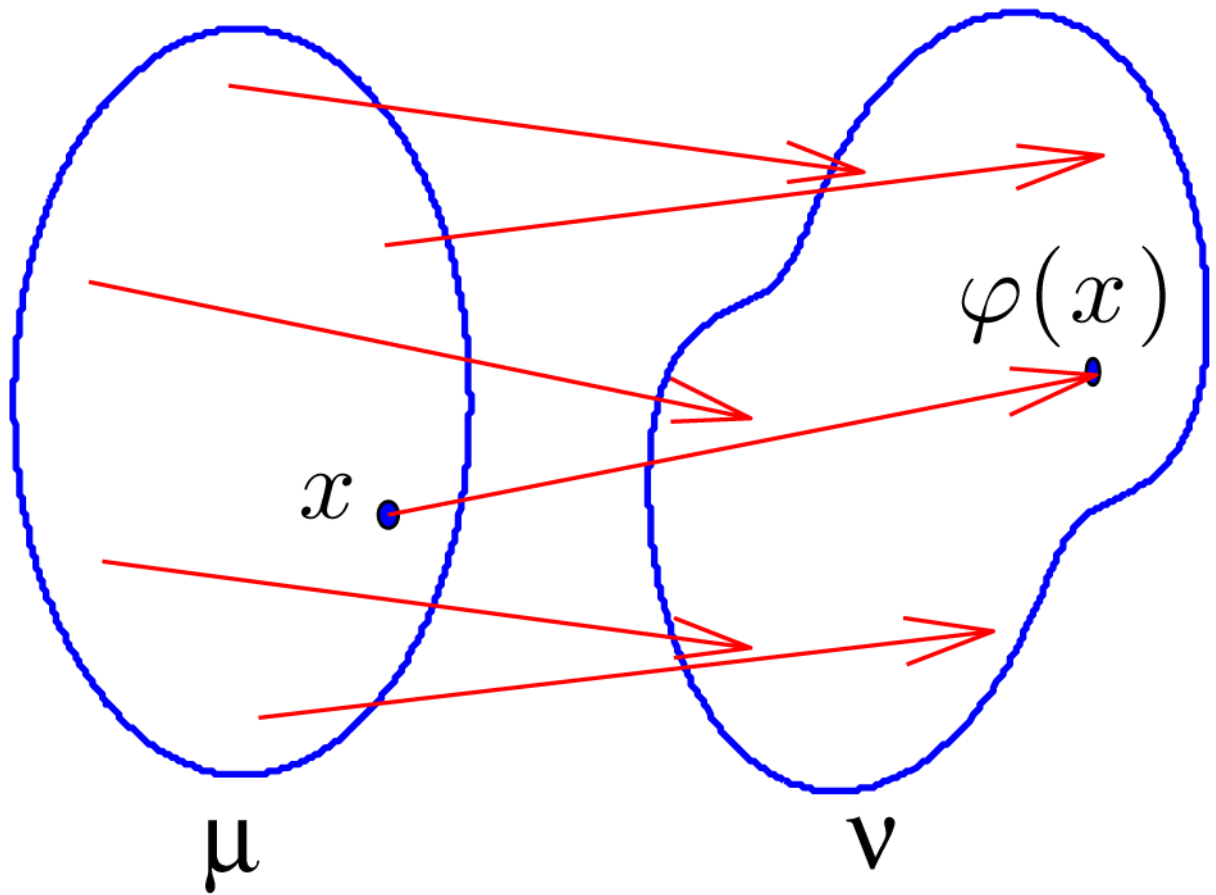
its center of mass exceed a certain threshold. To do so, we compute the transportation error  $err_i = d_W(\mu|V_i, \mu(V_i)\delta_{x_i})$  in cell  $i$ . Let  $err$  be the average transportation error. We add particles to all cells where  $err_i > 1.7err$ . More precisely if there are any cells where  $err_i > 1.7err$  then we split the cell with the largest error into two cells  $V_{i1}, V_{i2}$ , and then recompute the Voronoi diagram to determine the centers  $m_{i1} = \mu(V_{i1}), m_{i2} = \mu(V_{i2})$  of these two new cells. We repeat this splitting process until all the Voronoi cells have error less than  $1.7err_i$ . The choice of 1.7 was based on empirical evaluation with several datasets.

7. In order to reduce the number of particles in areas in each image where many particles are not needed, we merge nearby particles. Two particles are merged if, by merging them, the theoretical upper bound defined in (16) is lowered. When particles of mass  $m_1$  and  $m_2$  and locations  $x_1$  and  $x_2$  are merged to a single particle of mass  $m = m_1 + m_2$  located at the center of mass  $x = \frac{m_1x_1 + m_2x_2}{m_1 + m_2}$  the error in the  $m_1 + m_2$  distance squared is bounded by:

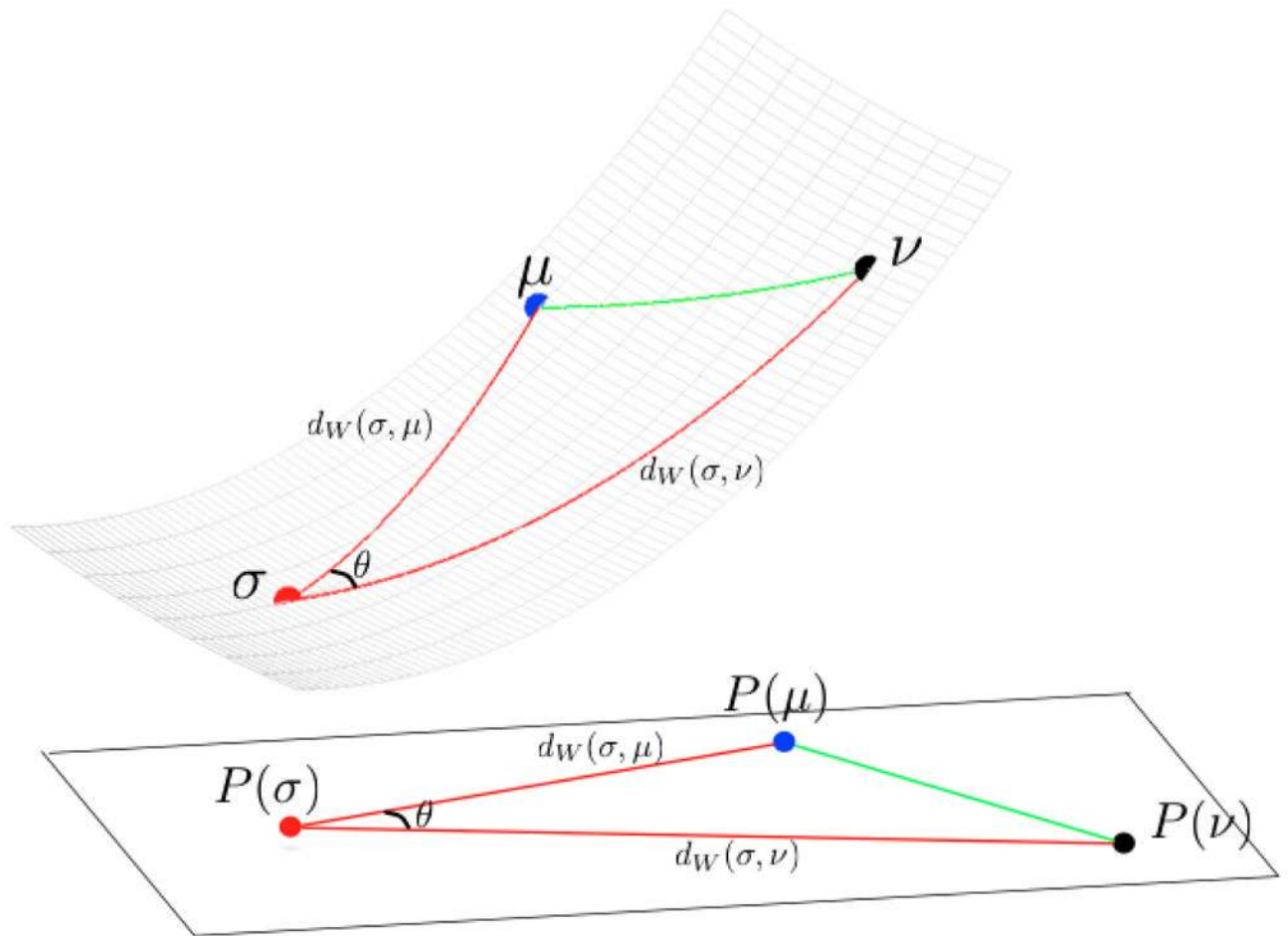
$$d_W^2(m_1\delta_{x_1} + m_2\delta_{x_2} + \mu_{rest}, m\delta_x + \mu_{rest}) \leq \frac{m_1m_2}{m_1 + m_2} |x_1 - x_2|^2$$

We merge the particles that add the smallest error first and then continue merging until  $d_W(\mu_N, \mu_{merged}) < 0.15d_W(\mu_N, \mu)$ . This has the overall effect of shifting the histogram (taken over all images) of the approximation error to the right, where the inequality is used in the sense of the available upper bound on the distance (Observation 3 and the estimate above) and not the actual distance. This ensures that the final error for each image after merging is around  $1.15d_W(\mu_N, \mu)$ .

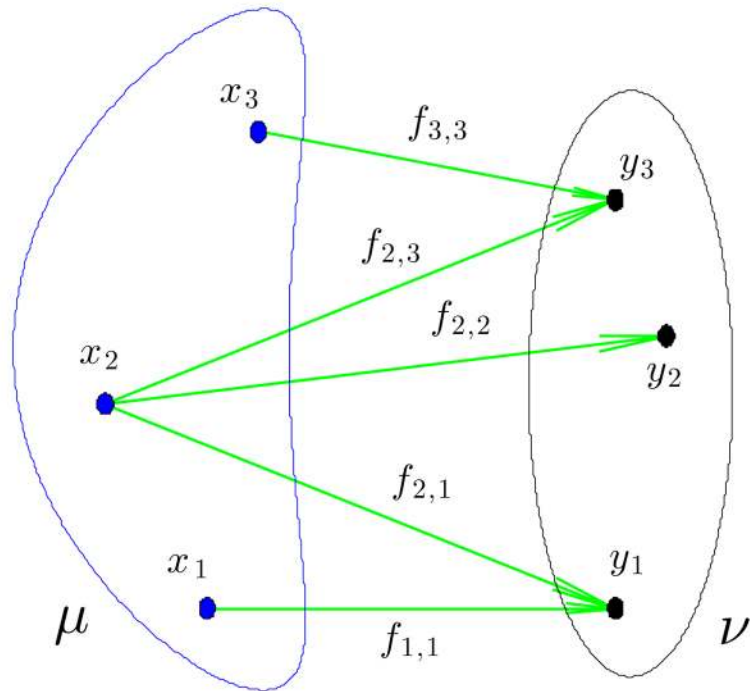
8. Finally, while the two steps above seek to find a good particle approximation for each image, with  $N$  particles or more, the final step is designed to make the particle approximation error more uniform for all images in a given dataset. For a set of images  $I_1, I_2, \dots, I_M$ , we estimate the average transportation error,  $E_{avg}$ , between each digital image and its approximation, as well as standard deviation  $t$  of the errors. We set  $E_{small} = E_{avg} - 0.5 * \tau$  and  $E_{big} = E_{avg} + 0.5 * \tau$ .
  - For images that have bigger error than  $E_{big}$ , we split the particles as in Step 6 until the error for those images are less than  $E_{big}$ .
  - For images that have smaller error than  $E_{small}$ , we merge nearby particles instead as in Step 7. The procedure we apply is to merge the particles that add the smallest error first and then continue merging until  $d_W(\mu, \mu_{merged}) = E_{small}$ .



**Fig. 1.** Optimal transport map  $\phi$  between measures  $\mu$  and  $\nu$  whose supports are outlined.

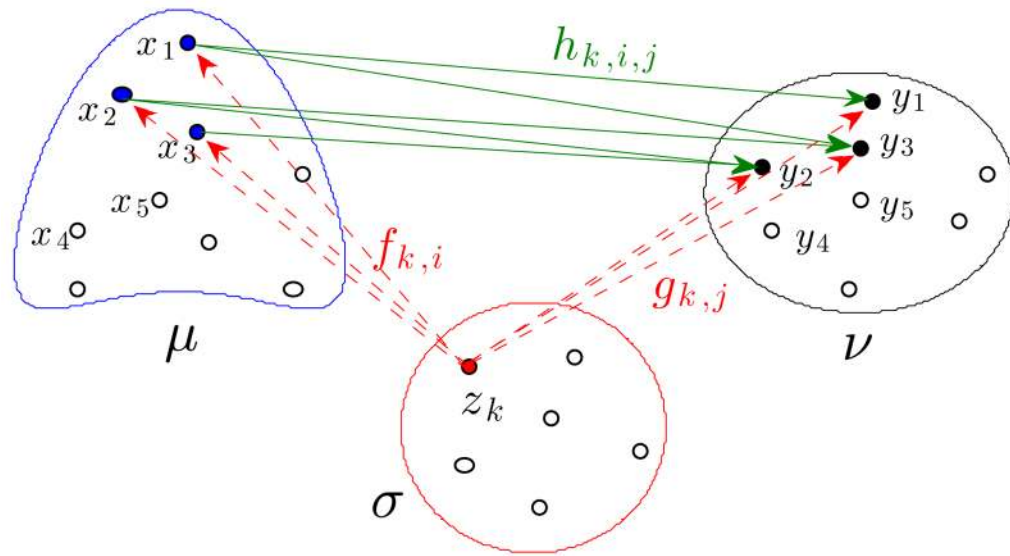


**Fig. 2.** The identification of the manifold with the tangent plane. Distances to  $\sigma$  as well as angles at  $\sigma$  are preserved.

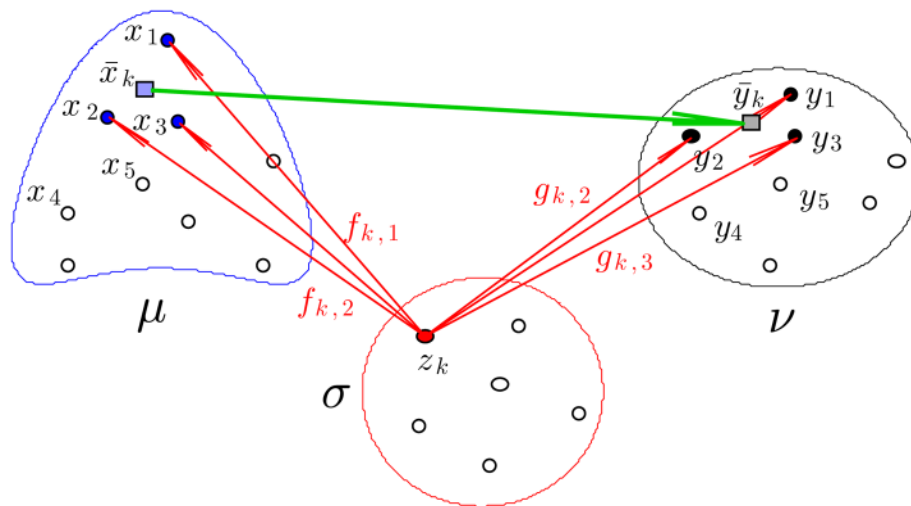


**Fig. 3.** When measures are discrete as in (4), then finding the transportation plan which minimizes the transportation cost, (5), necessitates splitting the particles. The arrows are drawn only for positive  $f_{i,j}$ .

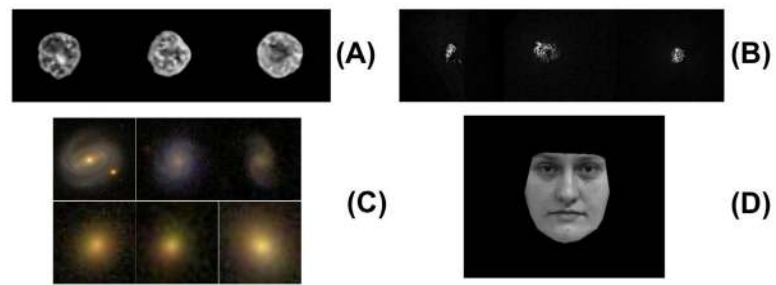




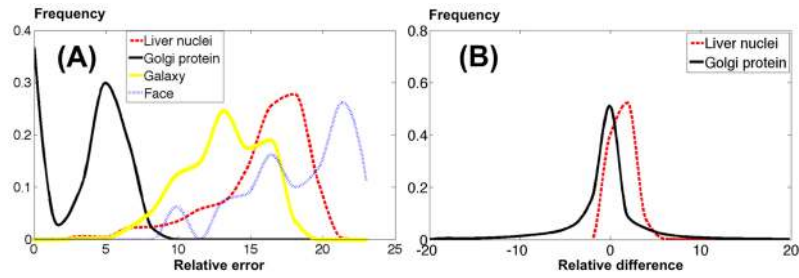
**Fig. 4.** Given  $f$  and  $g$  in (6), for each  $k$  fixed,  $h_{k,i,j}$  gives the optimal transportation plan between the ‘ $f$ -image’ and the ‘ $g$ -image’ of the particle at  $z_k$ . Arrows are drawn only for positive coefficients, and only for the particle at  $z_k$ .



**Fig. 5.** For the LOT distance we replace the full  $f$ -image and the  $g$ -image of the particle at  $z_k$  by their centers of mass  $\hat{x}_k$  and  $\hat{y}_k$ , (7). When there are many particles the images of particle at  $z_k$  tend to be concentrated on a few nearby particles. Thus the error introduced by replacing them by their center of mass is small.

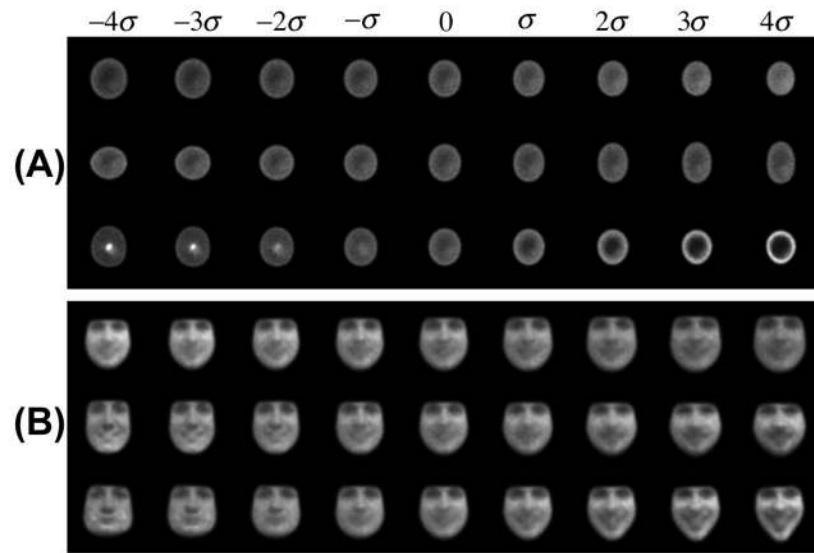


**Fig. 6.** Sample images for the two data sets. A: Sample images from the liver nuclear chromatin data set. B: Sample images from the Golgi protein data set. C: Sample images from galaxy image data set. D: Sample image from facial expression data set.

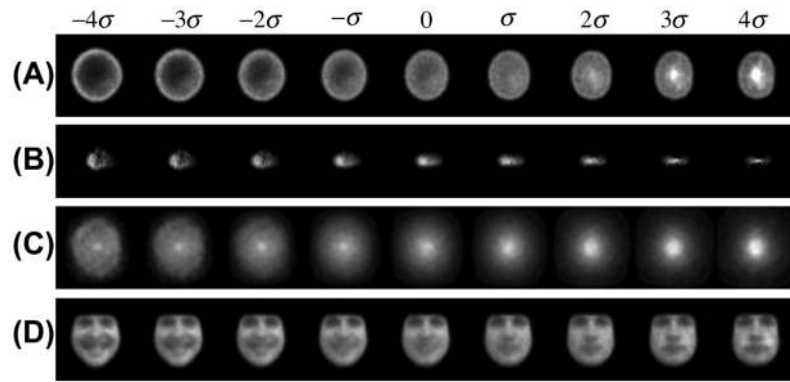


**Fig. 7.**

Histograms of particle approximation error and LOT error. A: The histogram of relative error generated during the particle approximation process. The x-axis represents the relative error in percentage, and the y-axis represents the portion of samples that has the corresponding relative error. B: The histogram of relative errors between LOT and OT for nuclei and golgi data sets. The x-axis represents the relative error in percentage, and the y-axis represents the portion of samples that has the corresponding relative error. The dotted line represents the relative errors for the liver nuclear chromatin dataset, and the solid line represents relative errors for the Golgi dataset.



**Fig. 8.** The modes of variation given by the PCA method combined with the LOT framework, for nuclear chromatin (A) and facial expression (B) datasets. Each row corresponds to a mode of variation, starting from the first PCA mode (top).



**Fig. 9.** Modes of discrimination computed using the penalized LDA method in combination with the LOT framework. Each row contains the mode of variation that best discriminates the two classes in each dataset. Parts (A),(B),(C),(D) refer to discrimination modes in nuclear morphology, golgi proteins, galaxy morphology, and facial expression datasets, respectively.

**Table 1**

Average classification accuracy in liver data

	<b>Feature</b>	<b>OT</b>	<b>LOT</b>	<b>EMD-L1</b>
Case 1	89%	87%	86%	84%
Case 2	92%	88%	89%	86%
Case 3	94%	91%	90%	87%
Case 4	80%	87%	85%	84%
Case 5	71%	76%	74%	76%
Average	85.2%	85.8%	84.8%	83.4%

**Table 2**

Average classification accuracy in golgi protein data

	OT			LOT		
	gia	gpp	gia	gpp	gia	gpp
gia	79.2%	20.8%	86.2%	13.8%	83.8%	16.2%
gpp	28.6%	71.4%	35.6%	64.3%	30.7%	69.3%



**Table 3**

Average classification accuracy for EMD-L1 in golgi protein data

	EMD-L1	
	gia	gpp
gia	85.8%	14.2%
gpp	31.3%	68.7%