# Nonnegative Matrix Factorization with Earth Mover's Distance Metric for Image Analysis

Roman Sandler, *student member, IEEE,* and Michael Lindenbaum, *member, IEEE*

**Abstract**—Nonnegative matrix factorization (NMF) approximates a given data matrix as a product of two low-rank nonnegative matrices, usually by minimizing the $L_2$ or the KL distance between the data matrix and the matrix product. This factorization was shown to be useful for several important computer vision applications. We propose here two new NMF algorithms that minimize the Earth mover's distance (EMD) error between the data and the matrix product. The algorithms (EMD NMF and bilateral EMD NMF) are iterative and based on linear programming methods. We prove their convergence, discuss their numerical difficulties, and propose efficient approximations. Naturally, the matrices obtained with EMD NMF are different from those obtained with $L_2$-NMF. We discuss these differences in the context of two challenging computer vision tasks, texture classification and face recognition, perform actual NMF based image segmentation for the first time, and demonstrate the advantages of the new methods with common benchmarks.

**Index Terms**—nonnegative matrix factorization, Earth mover's distance, image segmentation

## 1 Introduction

In computer vision we often need to learn characterizations of visual classes from examples. Examples representing class combinations are usually much easier to obtain than examples of a single class. Nonnegative matrix factorization (NMF) is a natural choice for learning the characterizations from such mixtures when the class characterization mixtures correspond to weighted sums of single class characterizations. Histograms and averaged feature vectors are some examples.

NMF is a representation of a nonnegative matrix as a product of two nonnegative matrices. The factorization becomes useful and interesting when the multiplied matrices are of low rank, implying usually that the factorization is approximate. In this case, the decomposition is useful for signal representation as an additive combination of a small number of atomic signals (part-based representation).

The factorization and the first algorithm for finding it were introduced by Paatero and Tapper [41]. An efficient multiplicative update algorithm was proposed by Lee and Seung [29], [30]. Different aspects of this latter algorithm were analyzed and many improvements were proposed [6], [17], [24], [23], [16], [60]. The NMF technique has been applied to many applications in the fields of object and face recognition, action recognition, and segmentation [60], [55], [48].

Consider a given descriptor $h_j^*$ which is a sum of several basic descriptors: $\vec{h}_j^* = \sum_i \vec{h}_i w_{ij}$. A set of descriptors, $H^* = (h_1^*| \ldots |h_m^*)$, may be written as $H^* = HW$, where the columns of $H$ are the basic descriptors and the columns of $W$ are the mixing weights. The basic algorithm proposed by Lee and Seung [30] gets a matrix $H^*$ and tries to find a pair of low rank nonnegative matrices $H$ and $W$ satisfying

$$\min_{H,W} Dist_\phi(H^*, HW) s.t. W \geq 0, H \geq 0, \qquad (1)$$

where the distance $\phi$ is either the Frobenius norm or the Kullback-Leibler distance. Although these distances have nice mathematical properties (e.g., bounded reconstruction error for the Frobenius norm [23]), they are not always the best choice for signal comparison. Some variations, adding a bias to desirable properties such as locality, were suggested [33], [26], [16]. The obvious nonuniqueness of the factorization was also discussed [17], [6] and usually resolved by some problem specific bias.

We believe that measuring the dissimilarity of $H^*$ and $HW$ as $L_2$ or KL distance, even with additional bias terms, is inappropriate given the nature of errors in realistic imagery, and other types of distances should be preferred. In particular, the Earth mover's distance (EMD) [45] metric is known (e.g., [58], [45], [32], [22]) to quantify the errors in image or histogram matching better than other metrics. The error mechanism, modeled as a complex local deformation of the original descriptor, is often a good model for the actual distortion processes in image formation. The EMD, whose variants are known as the Monge-Kantarovich problem, Wasserstein metric, Mallows distance, etc., was first applied to computer vision tasks by Werman et al. [58] and generalized by Rubner [45]. In this work we propose to factorize the given matrix using the EMD. That is, we consider here the minimization (1) where $\phi$ is the EMD metric.

- *R. Sandler and M. Lindenbaum are with the Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel, 32000.*

We propose here two NMF algorithms for the EMD metric, denoted EMD NMF and bilateral EMD NMF. Both differ notably from the multiplicative update algorithm [29] and its variations [6]. The EMD NMF algorithms are based on linear programming steps, and as such are more closely related to the techniques presented in [24].

We start by showing that, in principle, NMF may be used for image modeling. We then discuss a distortion model that motivates our use of the EMD metric. In the main part of the paper we propose the EMD based NMF and provide a linear programming based algorithm for the factorization. A more efficient algorithm, based on Wavelet EMD approximation [53], is described as well. Two EMD tasks are considered. The more general algorithm, denoted bilateral EMD NMF, is suitable for the case when the distortion is modeled well by small, in the EMD sense, errors in both spatial and feature domains. The simpler algorithm is preferred when the distortion fits the EMD model in only one of the domains.

We then examine the proposed factorizations with three vision tasks: texture modeling, face recognition, and image segmentation. Given an unlabeled image containing multiple textures, we extract the descriptors of individual textures using EMD NMF instead of actually segmenting the image. In the face recognition task, we handle unaligned facial images with some pose changes and different facial expressions. Finally, we show, for the first time, actual NMF based image segmentation. In all cases we consider sets of naturally deformed signal samples and reconstruct parts which appear to be the meaningful original signals. Our main contributions in this part of the paper are in demonstrating the superior performance of basic EMD NMF methods over other component analysis methods, some of which use problem specific bias terms.

This paper extends a preliminary version [50] by proposing the more general bilateral EMD NMF and discussing its relation to image modeling. With this as our foundation, we propose for the first time an NMF-based method for image segmentation.

This paper continues as follows: the relation of the NMF variables to the image model is shown in section 2. The formal definitions of NMF with the EMD metrics as well as the linear programming based algorithm are presented in section 3. Wavelet based approximation and other practical implementation details of the proposed factorization are discussed in section 4. Experiments with two actual vision tasks are discussed in section 6.

## 2 Observations and intuitions

The EMD NMF methods we propose are general and are not limited to an image domain. For concreteness and a more intuitive explanation, we chose to focus here on image representation. Consider an image $f(\vec{x})$ describing some feature $f$ as a function of the coordinate $\vec{x}$. We shall be interested in two types of histograms representing, respectively, parts of the image and parts of the feature space.

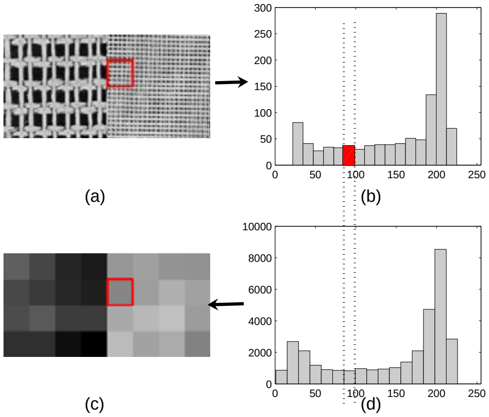


Fig. 1: Bilateral relation between the spatial and the feature domain representation. The image (a) may be represented in two domains. The highlighted spatial bin is associated with feature distribution $h_{x_0}(f)$ - (b). The highlighted feature bin in the whole image histogram $h(f)$ (d) is associated with spatial distribution $h_{f_0}(x)$ - (c). The highlighted bins in the spatial (c) and the feature (b) distributions are identical; $h_{f_0}(x_0) = h_{x_0}(f_0)$.

**A feature distribution** $h_{\vec{x}}(f)$ corresponds to a *region* $R_{\vec{x}}$ in the image and describes the feature distribution corresponding to the pixel values in this region.

**A spatial distribution** $h_f(\vec{x})$ corresponds to a *subset* $f$ of the feature space and describes the distribution of spatial locations corresponding to pixels having a value in this subset. Note that the spatial distributions do not necessarily sum to one.

See Figure 1 for the relations between the two domains and the respective histograms. In this work we consider only spatial regions and feature domain subsets large enough to contain a reasonable number of samples. Note that many other image representations follow this formulation. Two examples are orientation histograms [34] and Gabor jets [51]. While both coordinates may be multidimensional, e.g.,Gabor jets, we chose to discuss only a scalar feature $f$ in the following lines for simplicity.

Consider representing an image object, or several similar objects (denoted visual class through this paper), using spatial and feature distributions. Ideally, we would expect such an object to be associated with the same feature vector in all its locations. We would also expect the spatial distributions to be piecewise constant within the objects for every feature subset. Naturally, this expectation is unrealistic and the respective distributions are somewhat different, though these differences often follow a systematic pattern described below.

Consider a region belonging to a visual class with some ideal gray level histogram $h(f)$. Different regions of the same class may be associated with different surface normal directions and corresponding histograms which are brighter or darker. In this case, the absence of some gray level in the histogram is better explained by the presence of additional gray levels in nearby feature histogram bins than in the distant, unrelated bins. Consider now the spatial domain. In realistic textures, the distribution of gray levels in every region is not entirely uniform. Consider, for example, two adjacent regions in an image of a zebra. One region may contain more black pixels than the other, but the union of the regions has a histogram which is closer to the ideal class histogram. More generally, the absence of some gray level in a spatial bin is better explained by the presence of surplus instances of this gray level in nearby spatial bins than in other locations. This model of distortion leads to comparison of distributions with the Earth mover's distance, as will be explained in greater detail in the next section.

The proposed image model is well-suited to the NMF representation. Let the $(i,j)$-th element of $H^*$ measure the number of pixels with the $i$-th feature in the $j$-th region of the image. Then, the $j$-th column of $H^*$ contains the feature distribution in region $j$, $h_j(f)$. Analogously, the $i$-th row contains the spatial distribution of the $i$-th feature subset, $h_i(\vec{x})$. The factorization variables, $H$ and $W$, refer to the feature and spatial representations of the visual classes of the image. The columns of $H$ represent the ideal feature distributions and the rows of $W$ represent the ideal visual class locations, the image segments. The value of the $(i,j)$-th bin in the product matrix $HW$ is the sum of $i$-th feature probabilities in different classes weighted by their relative area in $j$-th region. In other words, it tells us how many of the feature values in the range $i$ we expect to find in region $j$, which is exactly the property the $(i,j)$-th bin of the matrix $H^*$ measures.

By factorizing $H^*$, we perform clustering in both spatial and feature domains. For image segmentation it is common to consider such groupings and gather pixels with similar appearance features and spatial locations. Some methods, for example, explicitly use this principle by clustering pixels in a combined (color, spatial coordinates) space [54], [14] Here we show that NMF models both the spatial and the feature image descriptors in a complementary way and acts as an iterative, EM-like, segmentation algorithm.

For reasonable factorization we should ensure that $H^* \approx HW$ and that the differences follow the local deformation model we discussed earlier. This compels us to require minimization of the EMD error between both the rows and the columns of $H^*$ and $HW$. In the next sections we quantify these requirements and use them to propose EMD NMF.

# 3 EMD NMF

Consider $M$ nonnegative histograms with $N$ bins. The histograms are represented in a matrix form, $H^* \in \mathfrak{R}^{N\times M}$, where the $j$-th histogram is the column $H^*_j$. The matrix $H^*$ may be decomposed into a product of $H \in \mathfrak{R}^{N\times K}$ and $W \in \mathfrak{R}^{K\times M}$, where $H$ and $W$ are interpreted as $K$ basis vectors in two complementary domains. In most cases, a low dimensional approximation is more meaningful than exact factorization. Then, the desired factorization $H,W$ is a solution of eq. (1) for small $K$ values. Let $Dist_\phi(A,B)$ be the sum of distances $\phi$ between the corresponding columns of $A$ and $B$. Then, $H^{*T} \approx W^T H^T$ implies that $Dist_\phi(H^*,HW)$ is the sum of distances between the feature histograms. Analogously, $H^{*T} \approx W^T H^T$ implies that $Dist_\phi(H^{*T},W^TH^T)$ is the sum of distances between the spatial histograms. Therefore, in order to find the spatial distributions, we should factorize $H^{*T}$ by solving

$$\arg\min_{H,W} Dist_\phi(H^{*T},W^TH^T) s.t. W \geq 0, H \geq 0. \quad (2)$$

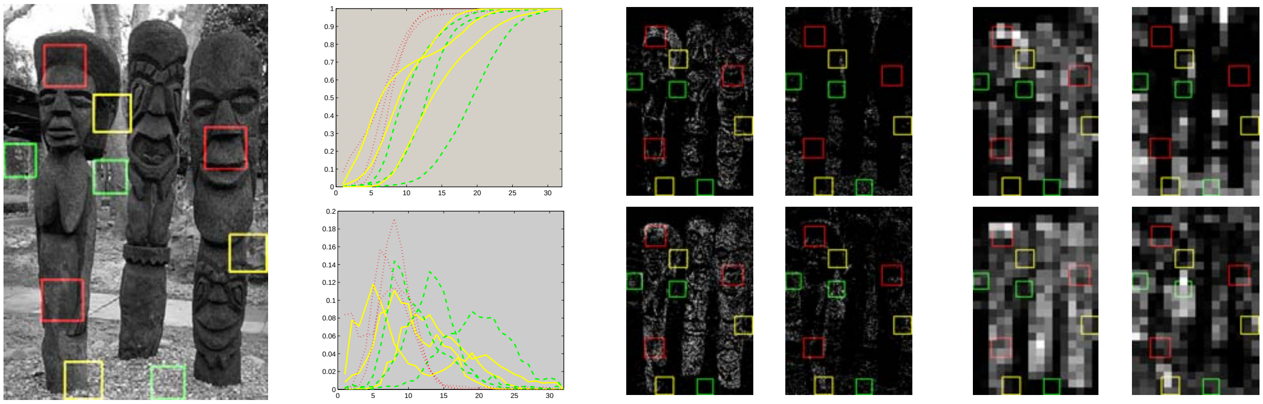
A joint clustering in both domains is, therefore,

$$\begin{aligned} \arg\min_{H,W} \quad & \lambda_1 Dist_\phi(H^*,HW) + \lambda_2 Dist_\phi(H^{*T},W^TH^T) \\ s.t. \quad & W \geq 0, H \geq 0. \end{aligned} \quad (3)$$

Conveniently, the $L_2$ distance is bin-wise and $Dist_\phi(H^*,HW) = Dist_\phi(H^{*T},W^TH^T)$. Thus, segmenting an image in spatial and feature domains is equivalent to solving the traditional $L_2$-NMF of the feature distribution matrix associated with this image. Unfortunately, this algorithm fails for real images. Solving (3) with $L_2$-NMF implicitly associates the error independence assumption with different histogram bins. This assumption is not a good model for the sample deviation in the approximation $H^* \approx HW$, neither in the feature nor the spatial domain. As already mentioned, we propose to use the EMD metric for column comparison and show its ability to solve such problems.

## 3.1 Earth mover's distance

The Earth mover's distance (EMD) evaluates the dissimilarity between two distributions in some feature space, where a distance measure between single features is given [45]. For image features, the EMD is motivated by the following intuitive observation: Some histogram bin mass may transfer to nearby bins due to natural image formation processes. The distance between two distributions which may be considered as small local deformations of each other should be less than that of other distribution pairs which differ in non-neighboring bins. Intuitively, we can view the traditional EMD metric as a sum of the changes required to transform one distribution into the other with low cost given to local deformations and high cost to nonlocal ones. Formally, the EMD



(a) The original image (b) Feature histograms (c) Feature indicator images (d) Spatial histograms

Fig. 2: Intuitive explanation of the model. The feature distributions in the graphs ((b)) (lower, feature histograms, upper cumulative histograms) are associated with the squares on the image ((a)). The red dotted lines refer to the red squares lying inside the totem segments, the green dashed lines refer to the green squares lying inside the background segments, and the yellow solid lines refer to the yellow squares intersecting the totem and background segments. The feature indicator images ((c)) show the pixels with equal feature values. The respective spatial histograms are shown in ((d)).

distance between two histograms is formulated as a linear program (5, 6) whose goal is to minimize the total flow $f(i,j)$ between the bins of the source histogram ($i$) and the bins ($j$) of the target histogram for a given inter-bin flow cost $d(i,j)$; see [45]. The cost parameter $d(i,j)$, denoted also the ground distance, specifies the inter-bin flow cost for each pair of source and target bins. EMD is a metric when $d(i,j)$ is a metric as well; thus, we consider here only this type of cost function and denote it the underlying metric.

We consider a nonnormalized distance

$$EMD(h^s, h^t) = \sum_{i,j} f(i,j)d(i,j), \tag{4}$$

where $f(i,j)$ is a solution of:

$$\begin{aligned} \min_f \quad & \sum_{i,j} f(i,j)d(i,j) && (5)\\ \text{s.t.} \quad & f(i,j) \geq 0, \\ & \sum_j f(i,j) \leq h_i^s, \\ & \sum_i f(i,j) \leq h_j^t, && (6)\\ & \sum_{i,j} f(i,j) = \min\left(\sum_i h_i^s, \sum_j h_j^t\right), \end{aligned}$$

because the total flow in our case is prespecified.

#### 3.1.1 *Earth mover's distance between matrices*

We define EMD between two matrices with $M$ columns as a sum of EMDs between each column in the source matrix and the corresponding column in the target matrix:

$$\|H^s - H^t\|_{EMD} = \sum_{m=1}^{M} EMD(H_m^s, H_m^t). \tag{7}$$

For columns representing feature vectors, this distance measures the sum of distances between respective feature pairs. Naturally, to consider EMD in spatial domain, we should find $\|{H^s}^T - {H^t}^T\|_{EMD}$.

### 3.2 Single domain LP-based EMD algorithm

The general NMF problem is nonconvex and has a unique solution only for limited cases [17]. However, if one of the variable matrices $H$ or $W$ is given, the problem becomes linear. Thus, by consecutively fixing either $H$ or $W$, one can find a local minimum for (1) by solving a sequence of convex tasks. This approach is also applicable to the case at hand by a simple reformulation of the EMD linear programming problem. As a result, the local minimum of EMD NMF is found by solving a sequence of linear programming tasks.

Consider $h^s = H_m^*$ and $h^t = (HW)_m$. Note that both vectors are normalized histograms and thus sum to one: $\sum_i h_i^s = \sum_j h_j^t = 1$; this constraint implies that the columns of $W$ sum to 1 as well. With these normalizations, the linear programming constraints associated with the EMD between $H_m^*$ and $HW_m$ (eq. 6) become

$$\begin{aligned} & f_m(i,j) \geq 0, \\ & \sum_j f_m(i,j) = H^*(i,m), && (8)\\ & \sum_i f_m(i,j) = \sum_k H(j,k)W(k,m). \end{aligned}$$

Note that the constraint $\sum_{i,j} f_m(i,j) = 1$ is satisfied automatically since $\sum_{i,j} f_m(i,j) = \sum_i H^*(i,m) = 1$.

Note also that if we know $H$, both $f_m(i,j)$ and the matrix $W$ minimizing it may be found as:

$$\arg\min_{f,W} \sum_m \sum_{i,j} f_m(i,j)d(i,j) \quad \text{s.t. (8)}. \tag{9}$$

Analogously, if we know $W$, we can find both $f_m(i,j)$ and the matrix $H$ minimizing it as:

$$\arg\min_{f,H} \sum_{m} \sum_{i,j} f_m(i,j) d(i,j) \quad \text{s.t. (8).} \qquad (10)$$

Thus, given some initial guess for $H$ or $W$, we can improve the solution by the following two-phase Algorithm 1.

**Algorithm 1** EMD NMF

```
Input: The objective matrix $H^* \in \Re^{N \times M}$ and an
    initial guess for the basis $H^0 \in \Re^{N \times K}$.
Find $W^0$ using (9).
$k = 0$
repeat
    k=k+1
    Find $H^k$ using (10).
    Find $W^k$ using (9).
until
    $\epsilon > \left| \|H^* - H^k W^k\|_{EMD} - \|H^* - H^{k-1} W^{k-1}\|_{EMD} \right|$
Output: $W^k$ and $H^k$.
```

For columns representing feature distributions, this algorithm finds a set of basic distributions ($H$) and the mixing weights ($W$) to construct the samples in $H^*$ from this set. For the spatial domain we factorize $H^{*T}$. This way we find a set of basic spatial distributions (rows of $W$) and the mixing weights ($H$) to construct the samples in $H^*$ from this set.

### 3.3 Convergence

*Theorem 1.1:* Algorithm 1 converges to a local minimum

*Proof:*

1) **Feasibility:** First note that Algorithm 1 is a sequence of LP processes. We should show that a feasible solution exists for every one of them. The minimization (9) gets a pair $H^*, H^k$ of normalized matrices. Any normalized matrix $W^k$ ensures that $\sum_i H^*_{mi} = \sum_j (HW)_{mj}$ and thus implies that a feasible solution exists. This follows from EMD being a transportation problem, which has a feasible solution when $\sum_i h^s_i = \sum_j h^t_j$ [25]. An identical argument shows the existence of a feasible solution for minimization (10).
2) Linear programming, by definition, minimizes the flow cost and, due to (7), minimizes $\|H^* - HW\|_{EMD}$. Thus, applying (10) finds globally optimal $H^k$ for a given $W^{k-1}$ and applying (9) finds globally optimal $W^k$ for a given $H^k$.
3) Since the objective in (10) and in (9) is the same, $\|H^* - H^k W^{k-1}\|_{EMD} \leq \|H^* - H^{k-1} W^{k-1}\|_{EMD}$ and $\|H^* - H^k W^k\|_{EMD} \leq \|H^* - H^k W^{k-1}\|_{EMD}$.
4) From the above it follows that every cycle of Algorithm 1 monotonically decreases the distance $\|H^* - H^k W^k\|_{EMD}$. This distance is lower-bounded, and therefore the algorithm converges (to a local minimum).

### 3.4 Bilateral EMD NMF

Algorithm 1 minimizes the EMD distance between the corresponding columns of a given matrix and a matrix product approximating it. Note, however, that in the general case specified by eq. (3), our goal is to minimize EMD distance both between the corresponding columns and the corresponding rows. W.l.g. we shall denote the columns as feature distributions and the rows as spatial distributions, as we did in section 2. The proposed bilateral NMF is a mathematically similar extension of Algorithm 1: while Algorithm 1 considers only the feature domain but regards the spatial histogram errors as independent, we now add the minimization of the EMD in the spatial domain to the optimization function. Thus the bilateral EMD distance is

$$\begin{aligned} BEMD(H^*, HW) &= \lambda_1 \sum_{m=1}^{M} EMD(h^*_m, Hw_m) \quad (11) \\ &+ \lambda_2 \sum_{f=1}^{F} EMD(H^{*T}_f, W^T H^T_f). \end{aligned}$$

Both EMD terms depend, of course, on the ground distance metric [45]. See the detailed specification below.

To minimize this proposed distance, we extend the EMD NMF technique of alternating convex minimizations. Thus, analogously to Algorithm 1, each step of the proposed minimization is a linear programming task, and a sequence of such tasks achieves a local minimum and provides estimates for $H$ and $W$.

The EMD between one column of $H^*_m$ and $Hw_m$ is:

$$\min_{f_m} \quad \sum_{i,j} f_m(i,j) d^f(i,j) \text{ s.t. (8)} \qquad (12)$$

where $f_m$ is a variable measuring the flow that we want to minimize between the histogram bins, and $d^f(i,j)$ is a ground distance measuring the cost of moving between the bins. In the new distance we need to minimize the flow $f_m$ between feature histogram bins while also minimizing the flow $f_s$ between spatial histogram bins. Thus, the new cost function is:

$$\min_{f_m, f_s, z_i} \sum_{m,i,j} f_m(i,j) d^f(i,j) + \sum_{s,x,y} f_s(x,y) d^x(u,v) \quad (13)$$

subject to the constraints (8) and the additional constraints on the spatial flow for the $i$-th rows in $H^*$ and

$HW$:

$$\begin{aligned}
f_s(u,v) &\geq 0, \\
\sum_v f_s(u,v) &\leq H^*(i,u), \\
\sum_u f_s(u,v) &\leq \sum_k H(i,k)W(k,v), \\
\sum_{u,v} f_s(u,v) &= \min\left(\sum_x H^*(i,u), \sum_{k,v} H(i,k)W(k,v)\right).
\end{aligned} \tag{14}$$

The ground distance $d^x(u,v)$ measures the cost of moving between the spatial bins $u$ and $v$.

The alternating steps are:

**W step** – minimize (13) for $f_m, f_s$, and $W$ such that (8) and (14).
**H step** – minimize (13) for $f_m, f_s$, and $H$ such that (8) and (14).

Note that the two sets of constraints (8) and (14) are not of the same form. The first specifies equality constraints and thus requires the total flow $\sum_i \sum_j f_m(i,j)$ to equal one. This is necessary to ensure that the columns of the solution matrices $H$ and $W$ still sum to one. The second constraint set (14), on the other hand, cannot be of the equality type, because formally there is no constraint on the sums of the $H$ and $W$ rows. In practice, however, the sums of the $HW$ rows are very similar to the sums of the $H^*$ rows. We apply here the standard inequality constraints of the EMD [45]. In a sense, this formulation of the problem may be regarded as solving EMD NMF between the columns with an EMD penalty term on the distance between the rows.

## 4 Efficient EMD NMF algorithms

It is possible to find a local minimum of (7) by iterative application of (10) and (9) starting from some reasonable guess for $H$. Linear programming is a well-studied problem and plenty of freeware and commercial solvers are available. However, for (10) the dimension of the problem is $MN^2$. This means that even for a traditional, relatively small problem of factorizing 100 facial images (each in $16 \times 16$ resolution), the LP optimization problem operates about 6 million variables. This makes even the specification of the problem (construction of the constraint matrix) a challenging task with today's solvers.

Most of the variables arise from the need to calculate the flow $f_m(i,j)$ (and possibly $f_s(i,j)$) in order to estimate the EMD between the histograms. The actual variables of interest are $H$ and $W$, which are only a small fraction of the variables in both (9) and (10).

### 4.1 A gradient based approach

The task of finding $H^k$ and $W^k$ in each step of Algorithm 1 is:

$$\begin{aligned}
H^k &= \arg\min_H \sum_m EMD(H^*_m, (HW^{k-1})_m) \\
W^k_m &= \arg\min_W EMD(H^*_m, (H^k W)_m).
\end{aligned} \tag{15}$$

For bilateral EMD NMF it is:

$$\begin{aligned}
H^k &= \arg\min_H BEMD(H^*, (HW^{k-1})) \\
W^k &= \arg\min_W BEMD(H^*, (H^k W)).
\end{aligned} \tag{16}$$

Given both $H$ and $W$, the error (7) can be calculated by solving $M$ (or $M+N$) independent, relatively small LP problems. We can solve both minimizations in (15) or (16) with some gradient based optimization over possible $H$ (or $W$) values. We are guaranteed to find the globally optimal solutions for each optimization because tasks (9) and (10) are convex.

Unfortunately, the complexity of a single precise EMD computation is $O(N^3 logN)$. Thus, the gradient based approach is expected to be complex as well.

### 4.2 A gradient optimization with WEMD approximation

Much effort has been devoted to speeding up the EMD calculation. For some underlying metrics it is easier than for others. For example, the match distance [58], which is the EMD between 1D histograms with a specific underlying metric, can be calculated as an $L_1$ distance between the cumulative versions of the histograms. A short survey of other methods suggested for faster EMD calculation may be found in [53], [42].

Shirdhonkar and Jacobs [53] proposed an efficient way to calculate the EMD between two histograms for some common underlying metrics $d(i,j)$. They proved that the result of optimization (5) is approximated very well by:

$$d(h^t, h^s)_{WEMD} = \sum_\lambda \alpha_\lambda |\mathtt{W}_\lambda(h^t - h^s)|, \tag{17}$$

where $\mathtt{W}_\lambda(h^t - h^s)$ are the wavelet transform coefficients of the $n$ dimensional difference $h^s - h^t$ for all shifts and scales $\lambda$, and $\alpha_\lambda$ are scale dependent coefficients. The different underlying metrics are characterized by the chosen scale weightings and wavelet kernels. Note that we are looking for local minima of some calculated EMD values and not for the EMD values themselves. Empirically we found that the local minima of EMD and WEMD are generally co-located, and thus the accuracy of the WEMD approximation of the actual EMD is less important for our goal.

Using the approximation (17) in (15) and (16) reduces the computational complexity of EMD to be linear. However, gradient methods naturally require knowledge of the gradient for the optimization variables. In the case of linear programming, the gradient may be derived from the solution of the dual problem; therefore, it is a byproduct of EMD calculation. Unfortunately, for the WEMD we need to calculate the gradient separately. This gradient is:

$$\nabla d_{WEMD} = \sum_{\lambda} \alpha_{\lambda} \cdot sign(\mathtt{W}_{\lambda}(h^t - h^s)) \cdot \nabla \mathtt{W}_{\lambda}(h^t), \quad (18)$$

where the explicit expression for the gradient $\nabla \mathtt{W}_{\lambda}(h^t)$, with respect to either W or H, is lengthy but straightforward. The complexity of the gradient (18) computation for $H$ is $O(N^2K)$. Note, however, that many additives remain constant between the iterations, and a smart calculation of the gradient greatly accelerated the computation.

Note that formally applying WEMD requires equality constraints in (14). This condition is not satisfied, but in practice the sums of the $H^*$ rows are similar to those of the $HW$ rows. Thus we used WEMD to find the EMD and its gradient for both the columns and the rows of the matrices.

### 4.3 The optimization process

We tested two optimization strategies: constrained optimization ($H \geq 0$, $W \geq 0$) of the distance (17), and unconstrained optimization with high penalty for negative variable values:

$$arg \min_{x} \sum_{m} d(H^*_m, HW_m)_{WEMD} + \Phi(x), \quad (19)$$

where $x$ is either $W$ or $H$ according to the relevant iteration and $\Phi(x)$ is a quadratic penalty term for $x < 0$. The latter unconstrained optimization appears to be more precise and faster.

Still, EMD NMF iterations are more complex than those of $L_2$-NMF. Using Matlab on an Intel Core 2 Quad 2.5 GHz processor, one full $H$ iteration for $M = 256, N = 32, K = 3$ (corresponding to the texture experiment described in section 5.2) takes around 30 seconds. One full $H$ iteration for $M = 200, N = 1024, K = 40$ (corresponding to the face recognition experiment described in section 5.1) may take up to 20 minutes.

## 5 Applications

NMF is useful especially when the analyzed data are a mixture of data from several sources. The use of the EMD metric is preferable over $L_2$ when bin dependent changes in histograms are more likely than independent ones. We start with a classic NMF application – face recognition. Interestingly, we found that the basis images obtained with EMD NMF differ considerably from the face parts supposedly obtained by $L_2$-NMF based methods, and lead to more accurate recognition. Then, we show how to extract texture descriptors from texture mosaic images. Finally, we use EMD NMF for image segmentation. In addition, we apply EMD NMF to estimate the quality of segmentation in precision / recall terms without supervision [49].

### 5.1 Face recognition

Face representation is a common test case for the NMF algorithms [29], [29], [33], [60]. Traditional NMF algorithms measure the differences between the faces with translation-sensitive $L_2$ related metrics, and thus require a good alignment between the facial features. It was shown that when the NMF is forced to prefer spatially limited basis components, these $L_2$ based algorithms perform better and provide perceptually reasonable parts [33], [26]. Here we show that the use of NMF with the EMD metric yields different, but still perceptually meaningful components. We found that these components are even more efficient for face classification.

#### 5.1.1 The EMD NMF components

Unlike the $L_2$ distance, the EMD is not very sensitive to small misalignments, facial expressions, and pose changes. The basis components provided by the EMD NMF are facial archetypes, each of which looks like a slightly deformed face. Each facial feature (e.g., the shape of the head, the haircut, or the shape of the nose) associated with some archetype is shared by several people. The face images in a set associated with the same person, and with different poses and expressions, are usually close (in the EMD sense) to a common facial prototype. This prototype is usually a convex combination of a small number of archetypes. Every face image is a combination of a few archetypes with relatively high coefficients (the prototype) and some other archetypes with much lower coefficients.

To better illustrate this structure, we start by considering a simple image set of 4 faces: two parents, their daughter, and another, male, non-family member (six images of each person; see examples in Figure 3). The people in the database share several features. The males have rougher facial features, while the female faces are smoother. The daughter shares facial features with both of her parents, especially with her father. The 24 images were put into the columns of $H^*$ and it was decomposed with EMD NMF with $k = 3$. The ground distance is the 2D distances between the image pixels. Note that the number of archetypes is smaller than the number of people. The resulting weight diagram is shown in Figure 3. The 3 weights associated with every image and the EMD NMF may be plotted in 2D because $w_1 + w_2 + w_3 = 1$. See Figure 3, where the input faces are plotted as $(w_1, w_2)$ points. The k=3 archetypes correspond to the $(1, 0)$, $(0, 1)$, and $(0, 0)$ points. The archetypes and some input images are shown as well. Note the similarity between the father (red circles) and the daughter (black triangles): both are represented mainly by the archetype in $(0, 0)$.

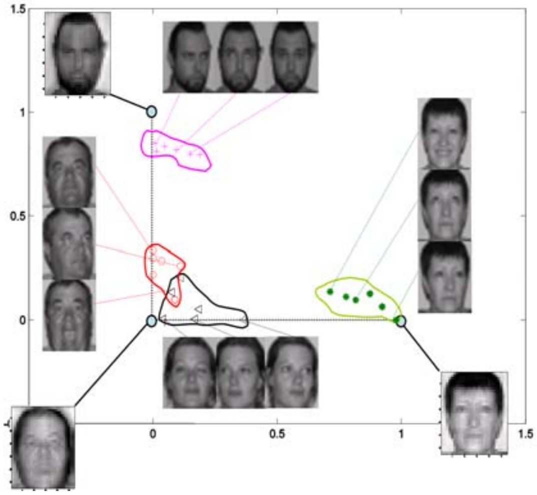


Fig. 3: Facial space for 4 people. The two-dimensional $(w_1, w_2)$ convex subspace is projected onto the triangle with corners in $(1, 0)$, $(0, 1)$, and $(0, 0)$. The corners of the triangle represent the basis facial archetypes obtained by EMD NMF. The inner points show the actual facial images weighted in this basis.

However, the father shares some male facial features with the archetype in $(0, 1)$. The daughter, on the other hand, shares many facial features with her mother's archetype, located in $(1, 0)$. The very noticeable changes in facial appearance caused by pose and expression are represented by small translations in the obtained subspace.

Interestingly, the representation of visual objects as a combination of object-like archetypes was suggested as a plausible model for object recognition in the human visual system [12], [57].

### *5.1.2 Face recognition algorithm*

To demonstrate the power of the EMD NMF, we use a straightforward recognition algorithm, based on 1-NN in the coefficient space. Let $\{(I_j, C_j)\ j = 1, \ldots, L\}$ be the training set ($I_j$ is an image, and $C_j$ is the corresponding class label).

```
Training:
Input: ${(I_j, C_j)\ j = 1, \ldots, L}$
  1: Normalize every image $I_j$ so that $\|I_j\|_1 = 1$.
  2: Decompose the matrix $I$ (with columns $I_j$), by EMD NMF, $I = HW$.
  3: Normalize every column $w_j$ so that $\|w_j\|_2 = 1$.
Output: $H$, $W$
Test:
Input: $I_t$, $H$, $W$.
  1: Normalize the test image $I_t$ so that $\|I_t\|_1 = 1$.
  2: Approximate $I_t$ as a convex combination of $H$'s columns, with weights
         $w_t = \arg\min_w EMD(I_t, Hw)$.
  3: Normalize $w_t$ so that $\|w_t\|_2 = 1$.
  4: Find $j^* = \arg\max_j < w_j, w_t >$.
Output: $C_{j^*}$.
```

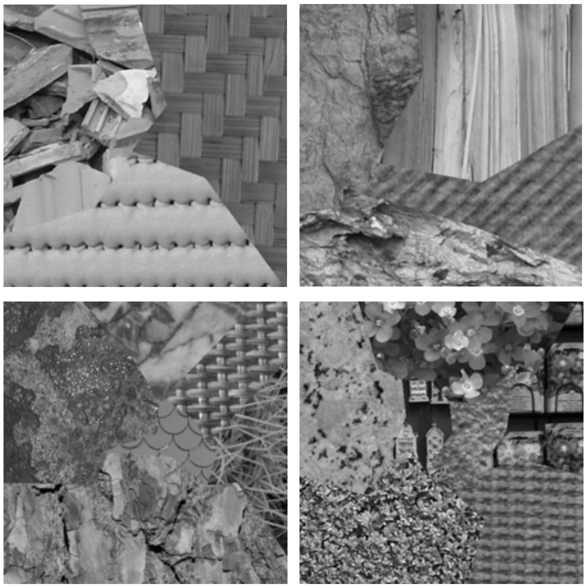


Fig. 4: Examples of texture mosaics. The mosaic borders change randomly, resulting in random combinations of the textures in the sample rectangles. Here, the images contain 3, 4, 6, and 7 textures. Note the high local variability of the textures.

This algorithm was successfully tested on two standard face recognition databases; see section 6.

## 5.2 Texture modeling

A texture mosaic is an image containing several types of textures in random arrangements; see examples from [38] in Figure 4. We consider the task of estimating the texture descriptors associated with each texture class of the mosaic. We also would like to classify the textures in each mosaic location, at least roughly (e.g., for consecutive segmentation). To that end, we consider the texture in nonoverlapping square image patches (blocks). The texture in each block is a positive mixture of the basic textures. Therefore the NMF suggests itself as an analysis tool.

The textures in the database [38] exhibit a lot of spatial variation. Even for relatively large blocks, the average texture descriptor in the block differs greatly from the average descriptor for the whole texture patch. Nor are the mosaics large enough to render descriptor distribution methods (e.g., [31]) effective. The EMD metric better compensates for the variability of the texture descriptor within the same texture than does $L_2$ [45], [11]. Therefore, EMD NMF is expected to be more accurate than $L_2$-NMF in estimation of the texture descriptors and the mixing coefficients thereof.

We rephrase the image model from section 2 as follows: Let each texture class be associated with some vector descriptor $h_k^{true}$ in each location of this texture. Then the $K$ descriptors associated with a mosaic image are $H^{true} = (h_1^{true}, \ldots, h_K^{true})$. Ideally, the mean texture descriptor in the $j$-th image block should be

$h^*_j = H^{true} w^{true}_j$, where $w^{true}_j$ is the vector of true fractions of the $j$-th block area associated with each texture class.

We applied the NMF to the texture mosaics by:

1) Converting the image to some feature vector representation. Following the findings in [45],we chose to work with the Gabor features, and thus each location is represented by a 6-orientation × 5-scale feature vector of Gabor responses [51]. Again, although the texture descriptors are organized in matrix columns, we consider 2D ground distance in the scale-orientation space.
2) Dividing the image into $M$ nonoverlapping rectangular blocks and calculating the mean feature vector $h^*_j$ for each block. We denote all the sampled mean block descriptors $H^* = (h^*_1|\ldots|h^*_M)$.
3) Finding the factorization $H^* \approx HW$. In this case only the domain of texture descriptors fits the EMD noise model, thus we use the single domain EMD NMF version.

The results of the factorization are the approximated representative texture descriptors $H = (h_1|\ldots|h_K)$ and the approximated fraction of each texture in each block $W = (w_1|\ldots|w_M)$. In section 6.2 we show that the results obtained with EMD NMF are more accurate and more robust than those obtained with $L_2$-NMF.

## 5.3 NMF and image segmentation

### 5.3.1 A naive NMF based segmentation algorithm

The NMF may be applied to image segmentation. We start by describing a preliminary, naive NMF based segmentation procedure and then continue developing it to achieve better results. Suppose that we use the NMF procedure to obtain an $H$ and $W$ associated with relatively small tiles $R_m$ covering the image. $W$ gives us a rough localization of the segments in the same resolution as the tiles; see Figure 5, top line. To obtain a refined, pixel resolution segmentation, we use the following Bayesian consideration: The $w_{k,m}$ fraction is the fraction of pixels coming from class $k$ in the tile $R_m$, and may be regarded as the prior probability that a pixel in $R_m$ belongs to the class $k$. We propose to decide, for every pixel, to which class it belongs, by means of a maximum a-posteriori decision. Suppose the image is scalar and $F(\vec{x})$ is the value in pixel $\vec{x}$. Let $H_{k,f}$ be the value of the bin associated with the feature value $f$ in the histogram of the class $k$. Then:

$$\begin{aligned} C(\vec{x}) &= \arg\max_k P(c_k|f = F(\vec{x})) \\ &= \arg\max_k \frac{w_{k,m} H_{k,F(\vec{x})}}{\sum_{k=1}^K w_{k,m} H_{k,F(\vec{x})}}. \end{aligned} \qquad (20)$$

The preliminary NMF-based segmentation algorithm is:

1) Tile the image with $M$ regions.
2) Compute $H^*$ for these regions.
3) Factorize $H^*$ with NMF and obtain $H$ and $W$.
4) Compute $C(\vec{x})$ for each image pixel using eq.(20).

For computational simplicity we use square tiles.

Unfortunately, this algorithm does not work well for real images. Even though the EMD NMF succeeds in finding reasonable approximations for $H$ and $W$ matrices, as shown in section 6, the inaccuracies in the obtained $W$ estimations cause frequent errors in the Bayesian assignment (20). Now we propose several improvements which bias the bilateral EMD NMF toward even more accurate $W$ estimation, and a corresponding better image segmentation algorithm.

### 5.3.2 Spatial smoothing

Recall that, ideally, the spatial basis histograms $W^T$ are piecewise constant. To use this information, we propose to implement the NMF under the BEMD distance with preference to minimizing the total variation [46]:

$$\begin{aligned} [\hat{H}, \hat{W}] &= \arg\min_{H,W} BEMD(H^*, HW) + \lambda TV(W), \\ &\text{where} \qquad (21) \\ TV(W) &= \sum_{k=1}^K \sum_{m=1}^M |d_x W_{m,k}| + |d_y W_{m,k}|. \qquad (22) \end{aligned}$$

$d_x W_{m,k}(d_y W_{m,k})$ is the difference between the spatial histogram value $W_{m,k}$ and another value $W_{m',k}$ associated with the following $x$ $(y)$ coordinate on the image plane.

In the new distance we need to minimize $z_x$ and $z_y$ – the differences between neighboring $W$ entries – in addition to minimizing the flows $f_m$ and $f_s$ between the feature and spatial histogram bins. Thus, the new cost function is:

$$\begin{aligned} \min_{f_m, f_s, z_i} \quad & \sum_{m,i,j} f_m(i,j) d^f(i,j) + \sum_{s,x,y} f_s(x,y) d^x(u,v) \\ & + \sum_{m,k} z_x(m,k) + z_y(m,k). \end{aligned} \qquad (23)$$

Subject to the constraints (8), (14), and the additional constraints on the spatial changes of $W$ (similar for the $x$ and $y$ directions):

$$\begin{aligned} z_x(m,k) &\geq 0 \\ -z_x(m,k) &\leq d_x(W_{m,k}) \leq z_x(m,k). \end{aligned} \qquad (24)$$

The ground distance $d^x(u,v)$ measures the cost of moving between the spatial bins $u$ and $v$.

The alternating steps become:

**W step** – minimize (23) for $f_m, f_s, z$, and $W$ such that (8), (14), and (24).

**H step** – minimize (23) for $f_m, f_s, z$, and $H$ such that (8), (14), and (24).

In practice, we use WEMD based optimization to solve each step, analogously to what is described in section 4.

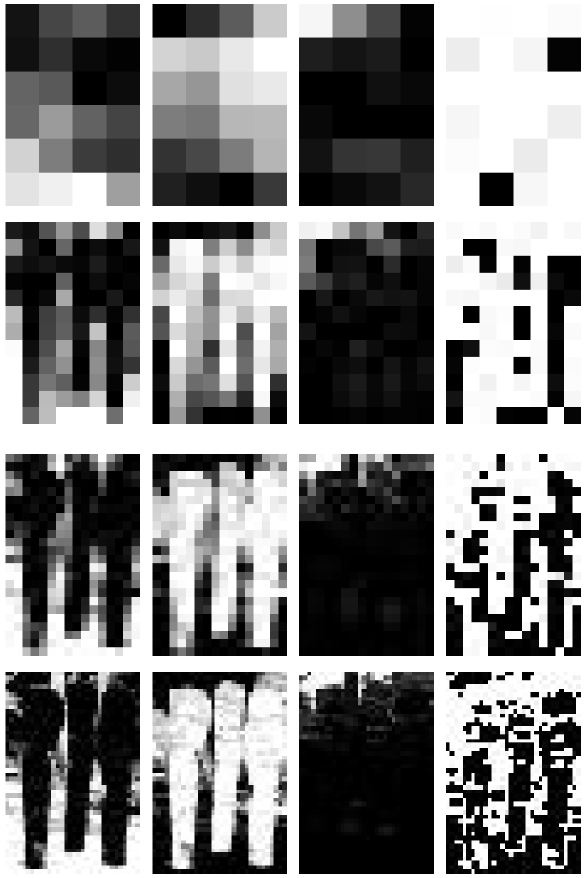


Fig. 5: $W$ estimates by multiscale BEMD. The results are for three-class factorization. The rightmost image for every scale shows the boundary class.

#### 5.3.3 Multiscale factorization

The preferred solution for $W$ is piecewise constant. Thus, we can save a lot of computational effort by working with $W$ in lower resolution during most of the factorization process. Moreover, the feature histogram estimation is more precise when applied to larger regions, e.g., see section 6.2. To use this twofold advantage we worked with a hierarchical, or multiscale, BEMD NMF solver.

First, the image is divided into large tiles and a small $H^*$ matrix is built. This matrix is factorized quickly and a rough $W$ along with a precise $H$ are estimated. Then, the new $H^*$ associated with smaller tiles is constructed and factorized with BEMD NMF. The latter factorization is initialized with the estimated $H$.

This process may be continued to finer resolutions; however, for the finer scales, the complexity grows and the model becomes less accurate. Therefore, we usually applied the factorization with 3-4 scales; see Figure 5.

#### 5.3.4 Boundary aware factorization

We refer to a boundary as a special, one pixel wide segment such that each pixel has at least a pair of neighbor pixels belonging to different object classes.

Because of its small size and high variability, the boundary is not modeled as a standard row of $W$. In each $W$ step the factorization algorithm associates a small part $\alpha$ ($2.5\%$ in our implementation) of each non-single-class region to be in the boundary segment; see Figure 5, the rightmost image for each scale. For a single-class region (i.e., a region with $W_{m,k} > 1-\alpha$ for some $k$) the boundary class weight is zero. The boundary class is usually associated with a wide distribution because of the high variation in the boundary feature values. Technically, the boundary class is associated with a column in $H$ and the $H$ step of BEMD NMF remains the same. Effectively we gain a twofold advantage: The boundary feature histogram effectively collects the feature distribution of the outliers in nonsingular regions and the class feature histograms become more precise.

#### 5.3.5 Bilateral EMD NMF segmentation algorithm

The final segmentation algorithm (Algorithm 2) is an enhancement of the first, naive algorithm proposed in the beginning of this section by the spatial smoothing term, the hierarchical decomposition, and the boundary extraction. The parameters are: $\beta_{max}$ is the number of scales (we used 3 or 4); $\Delta$ is the length of the tile side (we used $\sim 80$ pixels); $K$ is the manually specified number of classes.

Pixelwise Bayesian assignment sometimes creates a salt-and-pepper like mix between two classes if both classes have similar probability in a region. To avoid this kind of noise, we smoothed the obtained probability maps with several iterations of anisotropic diffusion.

**Algorithm 2** Bilateral EMD NMF segmentation

```
Input: $I(x,y)$, $K$, $\beta_{max}$, $\Delta$.
Guess initial $\hat{H} \in \Re^{n \times k+1}$ in a reasonable way. Set
  the boundary distribution as uniform.
for scale $\beta = 1 : \beta_{max}$ do
  Calculate $H^{*\beta}$ for $\frac{\Delta}{\beta} \times \frac{\Delta}{\beta}$ tiles.
  repeat
    Find $\hat{W}^{\beta}$ using $W$ step.
    $\hat{W}^{\beta} \leftarrow FindBoundary(\hat{W}^{\beta})$, see sec. 5.3.4
    Find $\hat{H}$ using $H$ step.
  until convergence
end for
Find $P(c_k|F(\vec{k}))$ with (20).
Smooth $P(c_k|F(\vec{k}))$ and find $C(\vec{x})$ with MAP.
Output: $\hat{W}^{\beta}$, $\hat{H}$, and $C(x,y)$.
```

## 6 Experiments

We now turn to test the performance of the proposed algorithms using standard benchmark databases for each application.

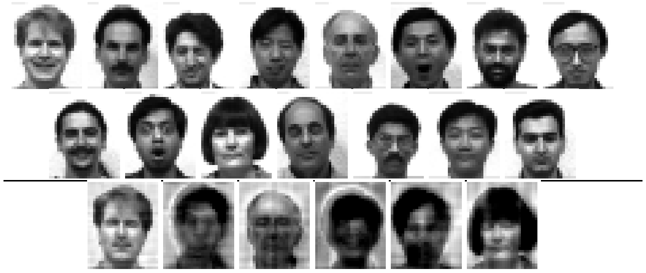
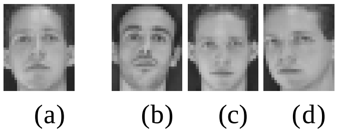


Fig. 6: The Yale faces database. The database contains images of 15 people, and we considered 8 images for each person. The first two rows show examples of the database images. The last row shows the basis images obtained with EMD NMF.



(a) (b) (c) (d)

Fig. 7: Typical recognition error in ORL database. When the test face image (a) is in a very different pose from that of the same person in the training set, the most similar person in the same pose (b) may be erroneously identified. The second-most similar identifications (c,d) are correct.

## 6.1 Face recognition experiment

We tested the EMD NMF based recognition algorithm on the popular Yale [7] and ORL [47] face databases. We follow the experimental procedure of [60], so that we can relate our results to those in [60] using the ORL database. Therefore, the face images are downsampled so that their longer side is 32 pixels. Moreover, as observed in [60], the recognition performance depends to a small extent on the partition of the database into the training and test sets. Following [60] and the approaches cited there, we provide the best results obtained in several training/test partitions.

In contrast to [60], we did not tightly align the faces by forcing the eye positions to coincide. Both databases contained images that were only roughly aligned. We did not touch the ORL database and, in the Yale database, we only centered the faces. This was necessary to avoid a situation in which facial position plays too great a role in identification.

The Yale face database contains fewer people than ORL, but is more challenging for recognition. We used a subset of it containing a set of images corresponding to the same lighting direction. Even with this restriction, the recognition task is not easy due to the high variability of expressions and to the possible presence of glasses. This implies that even for the best partition of the database into training and test sets, the test faces always differ considerably from their closest training examples. Four images were used to represent every person in the training set. A relatively high recognition rate of 86.6% was achieved using only 6 basis archetypes (representing 15 people). The archetypes obtained in this test are shown in Figure 6 together with examples of the faces they represent. Increasing the number of archetypes to 15 (one per person) increased the recognition rate to 95%. All the misses are due to glasses appearing in the test image but not in the corresponding training images.

It is interesting to observe that the proposed algorithm does not behave like a nearest neighbor algorithm with EMD metric. When a representative archetype for each person was computed as the image minimizing the sum of EMD distances over the corresponding training images, and 1-NN (with EMD metric) was used for recognition, accuracy was only $73.3\%$. This advantage of the EMD NMF based algorithm could be predicted also from the weight diagram in Figure 3, where, clearly, the father's images are closer to the daughter's mean image than to his own mean image (in weight space) and can be recognized only by the additional components.

The ORL database contains images of 40 people and is somewhat easier. As in [60], five images were used to represent every person in the training set. The recognition accuracy naturally changes with basis size $K$. For $K$ equal to or larger than the number of classes (people), the EMD NMF algorithm outperforms all the NMF based algorithms considered in [60], which often use much larger bases; see Table 1. Even with much lower basis dimension, the proposed algorithm achieves very high, competitive, accuracy.

Analyzing the (few) recognition errors, we found that they are associated with poses which differ notably from those in the training set; see Figure 7.

TABLE 1: Classification accuracies of different algorithms on the ORL database and the corresponding basis sizes cited from [60].

| Algorithm | NMF | LNMF | NGE | PCA | LDA | MFA |
|---|---|---|---|---|---|---|
| Basis Size | 158 | 130 | 121 | 105 | 39 | 48 |
| Accuracy (%) | 74.0 | 87.5 | **95.5** | 85.5 | 94.5 | **95.5** |

TABLE 2: Classification accuracy of EMD NMF on the ORL database for different basis sizes.

| Basis Size | 2 | 5 | 8 | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|
| Accuracy (%) | 8.5 | 70.5 | 87.5 | 94.5 | 90.5 | 95.0 | 96.5 | **97.0** |

## 6.2 Texture descriptor estimation

We applied the algorithm described in section 5.2 to 90 online generated mosaics [38]. Each test was repeated for combinations of two parameters: the number of textures in the mosaic ($K = 3, \ldots, 12$ textures) and the number of blocks $M = 16, 64, 256, 1024$ (number of columns in $H^*$). The blocks tessellate the image. Therefore, $M$ also specifies the block size to be $128 \times 128$, $64 \times 64$, $32 \times 32$, and $16 \times 16$ pixels respectively. In each test the $K$ parameter was set to the number of texture classes in the image.

We compared the estimated $H$ and $W$ matrices with the actual matrices $H^{true}$ and $W^{true}$ using the

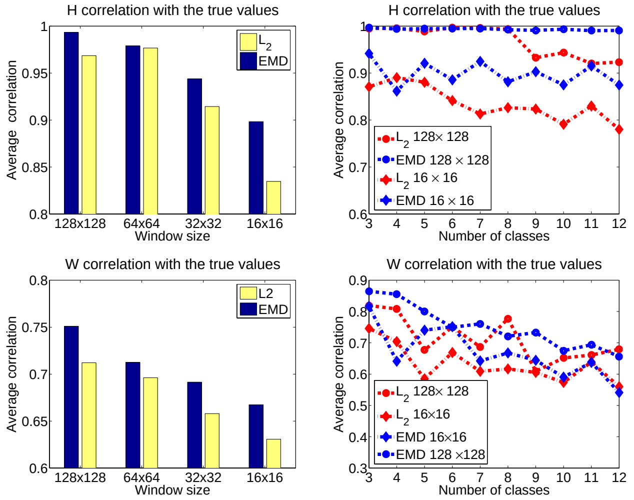


Fig. 8: Texture descriptor estimation accuracy. The first row shows the reconstruction quality of the basis descriptors and the second row shows the reconstruction quality of the mixing coefficients. The left column shows the average (over different $K$-s) reconstruction quality for the different sizes of the sampling blocks and the right column demonstrates the reconstruction quality as a function of the number of texture classes for two sizes of the sampling blocks.

following correlation measure:

$$Q_a(A, A^{true}) = \frac{1}{K}\sum_{i=1}^{K} \frac{< \vec{a}_i, \vec{a^{true}}_i >}{\|\vec{a}_i\|\|\vec{a^{true}}_i\|}. \qquad (25)$$

The estimated $Q_h = q(H, H^{true})$ and $Q_w = q(W^T, (W^{true})^T)$ values for the different test parameters are shown in Figure 8. The columns/rows are assigned to the respective ones in the true matrices by sequential greedy assignment, which maximizes $Q_a$. Note that as the block size increases, the descriptors $H^*$ are evaluated over a bigger area and are thus more precise for both metrics.

The graphs in Figure 8 illuminate two important differences in the behavior of the two metrics. Although they perform comparably when sufficient (64) samples of relatively reliable ($64 \times 64$ blocks) of data are available, EMD NMF outperforms $L_2$-NMF when the number of sample vectors is small or the samples less reliable. For the EMD metric, the performance of the H reconstruction does not depend on the number of classes, whereas for the $L_2$ metric it decreases with a larger $K$. These findings also support the observation that EMD is more robust when ideal data is not available.

In addition to the mean of the column/row correlations (25), we also measured their standard deviation. We found that the EMD NMF is generally associated with much smaller (in 30-50%) standard deviation than the $L_2$-NMF. The intuitive explanation is that while the $L_2$-NMF estimations of $H$ columns and $W$ rows are either very accurate or very inaccurate, the EMD NMF estimations are generally more stable. Together with the average correlation results, this makes the EMD NMF estimations for both $H$ and $W$ more reliable than those of $L_2$-NMF.

### 6.3 Segmentation

We experimented on two popular image databases: the Berkeley Segmentation Dataset [37] and the Weizmann Segmentation Evaluation Database [1]. Both databases are built on similar ideas and provide tools to benchmark algorithm performance using the manual segmentations of the database images. Both test performance in similar terms – an algorithm receives an $F$-number score for each segmented database image.

The evaluation task associated with the $F$-value score is different for the two databases. The score in the Berkeley database judges the algorithm by its ability to detect all object boundaries specified in manual segmentations and to avoid boundary detection in other places. The evaluation task in the Weizmann database is to specify the main object's pixels in the image as accurately as possible.

We performed a similar simple test on both databases. Each pixel was characterized with gray level value and gradient size as its 2D feature. Each image was segmented with Algorithm 2 into a manually specified number (between 2 and 7) of classes and the boundary class. In both tests the proposed algorithm showed consistent results; see some examples in Figures 9 and 10. However, the interpretation of these results is different for the two databases.

**Weizmann database.** The goal of this database benchmark is to detect the main object in the image accurately. The database was purposely designed to contain "a variety of images with objects that differ from their surroundings by either intensity, texture, or other low level cues." These low level cues may differ along the goal object as well as along the background. The images in the database are gray scale. The best achieved performance of the algorithm on this database was $F = 0.83$. According to [1], this performance is much better than that of N-Cut ($F = 0.72$) and MeanShift ($F = 0.57$) and even better than that of some complex multifeature algorithms. The algorithm best succeeded with images having different feature descriptions of the object and the background, no matter how complex this description is, and failed mostly on the images where the object and background descriptions share a large part of the feature space, especially if these shared features have large spatial presence; see examples in Figure 9.

**Berkeley database.** The Berkeley test checks an algorithm's performance on boundary detection tasks for color images. The ground truth segmentations include some of the image objects chosen manually. Algorithm 2 provides for each image point a probability to be a boundary point. These probability maps were tested by the database benchmark tools.

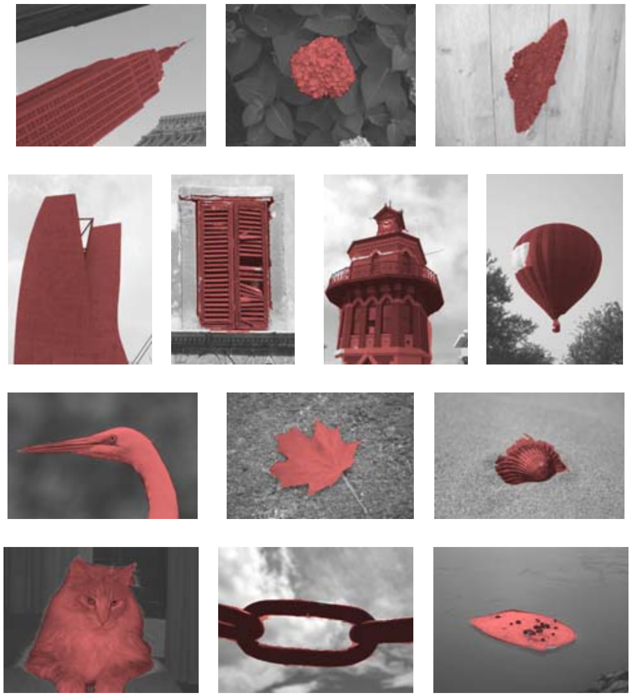


Fig. 9: Segmentation examples, Weizmann database

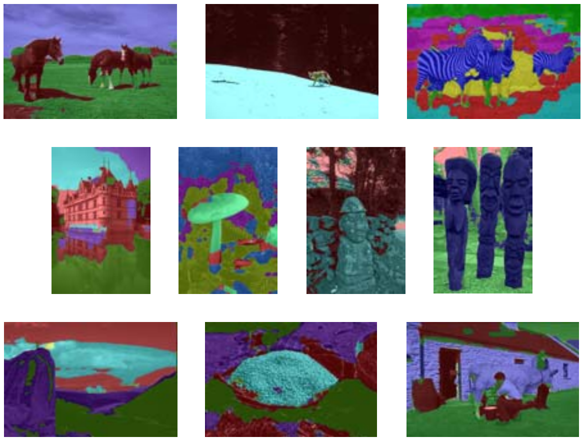


Fig. 10: Segmentation examples, Berkeley database

Testing the object on this database reveals both the merits and the deficiencies of the algorithm. While its results ($F = 0.55$) are worse than those obtained by state-of-the-art learning based algorithms [3], it should be noted that the state-of-the-art results are obtained using the color information from the images. Our results are similar to those obtained by N-cut and mean shift on grayscale images [49]. Looking at some examples, it is apparent that the algorithm is able to extract the appearance model but fails to exploit this knowledge to segment the fine details of the object. Stronger features (e.g., texture and color) and a more sophisticated final segmentation stage are needed to exhibit the strength of the proposed algorithm in this test.

## 7 Conclusions

A new type of NMF task, NMF with EMD metric, is proposed. The problem is solved with a linear programming based iterative algorithm. A WEMD [53] based optimization technique is proposed for fast implementation of the proposed algorithm. Algorithms based on the proposed EMD NMF outperformed previous NMF based algorithms in the context of two challenging computer vision tasks.

The main advantage of the new approach would seem to be its enhanced robustness. Consider, for example, the task of identifying a set of basis descriptors from mixture measurements. When the given measurements closely approximate linear combinations of the hidden descriptors, then the $L_2$ NMF technique suffices to accurately extract the basis. When the mixtures are, however, mixtures of deformed descriptors, this is no longer the case. Nonetheless, the deformed descriptors may be close, in the EMD sense, to the original descriptors. Then, the mixture of deformed descriptors is EMD close to the mixture of original descriptors (with the same weights). This lower sensitivity to deformations allows the EMD NMF to succeed when the $L_2$-NMF does not. Note that this situation is typical when we approximate a histogram from a small sample mixture.

The image model discussed in the paper proposes to use the enhanced properties of EMD NMF for a simple and elegant image description as a matrix product. Naturally, the simple linear model merely replaces the complex, nonlinear approximation with the more complex EMD metric. However, it allows an elegant image analysis independent of technical details.

Each of the considered applications is just a straightforward demonstration of the advantages of EMD NMF. Future research will be concerned with converting each of them into full-scale face recognition, database search, and segmentation tools.

## Acknowledgments

This work was supported by the Israeli Science Foundation. We thank Dr. Michael Zibulevsky and Dr. Boris Bachelis for valuable discussions.

## References

[1] S. Alpert, M. Galun, R. Basri, and A. Brandt". "Image segmentation by probabilistic bottom-up aggregation and cue integration.". In *CVPR*, "June" 2007.

[2] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *PAMI*, 20(2):168–185, 1998.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[4] M. Andreetto, L. Zelnik-Manor, and P. Perona. Non-parametric probabilistic image segmentation. In *ICCV*, 2007.

[5] O. Ben-Shahar and S. Zucker. The perceptual organization of texture flow: A contextual inference approach. *PAMI*, 25(4):401–417, 2003.

[6] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173, September 2007.

[7] P. N. Bellhumer, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 17(7):711–720, 1997.

[8] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *CVPRW*, 2004.

[9] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recogn. Lett.*, 19(8):741–747, 1998.

[10] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001.

[11] R. E. Broadhurst. Statistical estimation of histogram variation for texture classification. In *Texture Analysis and Synthesis Workshop, ICCV*, 2005.

[12] H. H. Bülthoff and S. Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. In *PNAS*, volume 89, pages 60–64, January 1992.

[13] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche. Unsupervised evaluation of image segmentation application to multi-spectral images. In *ICPR*, 2004.

[14] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, May 2002.

[15] T. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *CVPR*, 2005.

[16] I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *NIPS*, volume 18, pages 283–290, 2006.

[17] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts. In *NIPS*, 2003.

[18] J. H. Elder and R. M. Goldberg. Ecological statistics of Gestalt laws for the perceptual organization of contours. *J. Vis.*, 2(4):324–353, 8 2002.

[19] E. A. Engbers, M. Lindenbaum, and A. W. M. Smeulders. An information-based measure for grouping quality. *ECCV*, pages 392–404, 2004.

[20] F. J. Estrada and A. D. Jepson. Benchmarking image segmentation algorithms. In *IJCV*, 85(2):167–181, 2009.

[21] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[22] K. Grauman and T. Darrel. Fast contour matching using approximate Earth mover's distance. In *CVPR*, 2004.

[23] T. Hazan and A. Shashua. Analysis of l2-loss for probabilistically valid factorizations under general additive noise. Technical Report 2007-13, The Hebrew University, 2007.

[24] M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *J. Mach. Learn. Res.*, 7:1385–1407, 2006.

[25] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill Science / Engineering / Math, 2005.

[26] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, 2004.

[27] D. Jacobs. Robust and efficient detection of salient convex groups. *PAMI*, 18(1):23–37, 1996.

[28] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[29] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[30] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *NIPS*, 13:556–562, 2001.

[31] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, June 2001.

[32] E. Levina and P. Bickel. The Earth mover's distance is the Mallows distance: some insights from statistics. In *ICCV*, 2001.

[33] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *CVPR*, 2001.

[34] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *IJCV*, 60(2):91–110, 2004.

[35] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, 2001.

[36] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, May 2004.

[37] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[38] S. Mikeš and M. Haindl. Prague texture segmentation data generator and benchmark, 2006.

[39] P. Meer, B. Matei, and K. Cho. Input guided performance evaluation, in *Performance Characterization in Computer Vision*, pages 115–124. Kluwer, Amsterdam, 2000.

[40] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, XLII:577–685, 1989.

[41] P. Paatero and U. Tapper. Positive matrix factorization: a nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[42] O. Pele and M. Werman. Fast and robust Earth mover's distances. In *ICCV*, 2009.

[43] A. Rabinovich, T. Lange, J. Buhmann, and S. Belongie. Model order selection and cue combination for image segmentation. In *CVPR*, 2006.

[44] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[45] Y. Rubner. *Perceptual Metrics for Image Database Navigation*. PhD thesis, Stanford University, 1999.

[46] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[47] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, December 1994. IEEE.

[48] R. Sandler and M. Lindenbaum. Unsupervised estimation of segmentation quality using nonnegative factorization. In *CVPR*, 2008.

[49] R. Sandler and M. Lindenbaum. Unsupervised estimation of segmentation quality using nonnegative factorization. Submitted.

[50] R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth movers̀ distance metric. In *CVPR*, 2009.

[51] R. Sandler and M. Lindenbaum. Optimizing Gabor filter design for texture edge detection and classification. In *IJCV*, 84(3): 308-324 (2009).

[52] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, 1997.

[53] S. Shirdhonkar and D. Jacobs. Approximate Earth mover's distance in linear time. In *CVPR*, 2008.

[54] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. In *TIP*, 7(3:310–318, 1998.

[55] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008.

[56] S. Warfield, K. Zou, and W. Wells, III. Simultaneous truth and performance level estimation (staple): An algorithm for the validation of image segmentation. *MedImg*, 23(7):903–921, July 2004.

[57] S. Ullman. *High-level Vision: Object Recognition and Visual Cognition*. The MIT Press, Cambridge, MA, 1996.

[58] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. In *CVGIP*, volume 32, pages 328–336, 1985.

[59] L. R. Williams and K. K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *IJCV*, 34(2-3):81–96, 1999.

[60] J. Yang, S. Yang, Y. Fu, X. Li, and T. Huang. Non-negative graph embedding. In *CVPR*, 2008.

[61] Y. Yitzhaky and E. Peli. A method for objective edge detection evaluation and detector parameter selection. *PAMI*, 25(8):1027–1033, August 2003.

[62] H. Zhang, S. Cholleti, S. A. Goldman, and J. E. Fritts. Meta-evaluation of image segmentation using machine learning. In *CVPR*, 2006.

[63] Y. Zhang. A review of recent evaluation methods for image segmentation. *ISSPA*, 1:148–15, August 2001.