Barycenters of Natural Images – Constrained Wasserstein Barycenters for Image Morphing

Dror Simon
Technion
Haifa, Israel
dror.simon@cs.technion.ac.il

Aviad Aberdam
Technion
Haifa, Israel
aaberdam@cs.technion.ac.il

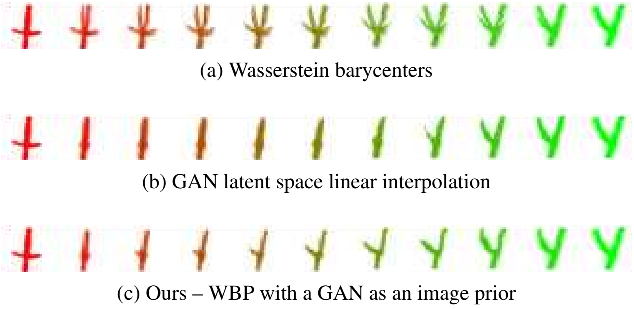
Abstract

Image interpolation, or image morphing, refers to a visual transition between two (or more) input images. For such a transition to look visually appealing, its desirable properties are (i) to be smooth; (ii) to apply the minimal required change in the image; and (iii) to seem "real", avoiding unnatural artifacts in each image in the transition. To obtain a smooth and straightforward transition, one may adopt the well-known Wasserstein Barycenter Problem (WBP). While this approach guarantees minimal changes under the Wasserstein metric, the resulting images might seem unnatural. In this work, we propose a novel approach for image morphing that possesses all three desired properties. To this end, we define a constrained variant of the WBP that enforces the intermediate images to satisfy an image prior. We describe an algorithm that solves this problem and demonstrate it using the sparse prior and generative adversarial networks.[1]

1. Introduction

Image morphing of two input images is a visual effect in which a sequence of images is obtained, transforming one image into the other. By denoting the input images as $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$, the objective is to find a sequence of N images $\{\boldsymbol{y}_i\}_{i=1}^N$, $\boldsymbol{y}_i \in \mathbb{R}^n$ that transform $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$. Generally, there are infinite possible ways of transforming one image into the other. Nevertheless, a pleasant transition should uphold the following properties. First, the difference between any two consecutive frames should be quite similar, leading to a *smooth* steady-paced animation. Second, the overall variation in the entire transition should be *minimal*, avoiding unnecessary changes.

The naive solution to consider for image morphing is a simple linear interpolation between the two images, i.e. $\boldsymbol{y}_i = \frac{N+1-i}{N+1}\boldsymbol{x}_1 + \frac{i}{N+1}\boldsymbol{x}_2$. While this method indeed produces a smooth transition, it leads to unnatural intermediate samples that contain unpleasant double-exposure artifacts. Therefore, to obtain a pleasant transition, an additional requirement is needed.


(a) Wasserstein barycenters

(b) GAN latent space linear interpolation

(c) Ours – WBP with a GAN as an image prior

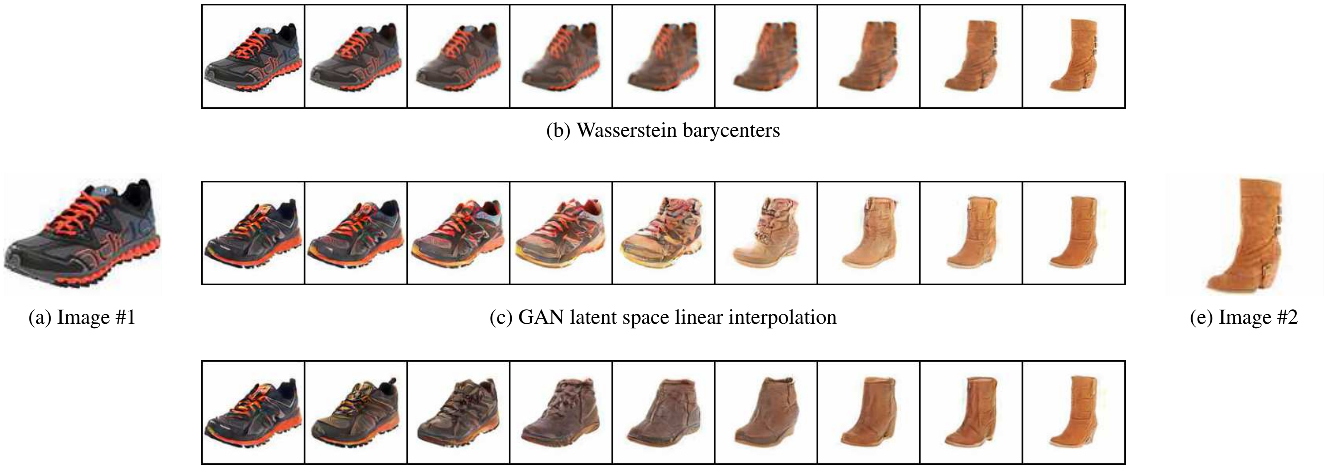
Figure 1: Morphing a 't' image to a 'y' using 3 different methods, where $\alpha \in \{0, 0.1, ..., 1\}$ (colors are used to emphasize the transition). In Figure 1a the intermediate images do not look like English letters. In Figure 1b the rate of the changes varies throughout the transformation. Figure 1c demonstrates a smooth transition of English characters. Images taken from the EMNIST dataset [1].

An approach that overcomes the double-exposure artifact, is solving the Wasserstein Barycenter Problem (WBP) [3, 4]. The Wasserstein barycenter is the probability distribution function that minimizes the mean of its Wasserstein distances [5] to each element in a given set of probability distributions. Considering two input probability distributions located on the simplex $\{\boldsymbol{p}_1, \boldsymbol{p}_2\} \in \Sigma_n$ the WBP is then defined as

$$\boldsymbol{p}_\alpha = \underset{\boldsymbol{q} \in \Sigma_n}{\arg\min} \; (1-\alpha)\mathcal{W}_2^2(\boldsymbol{p}_1, \boldsymbol{q}) + \alpha \mathcal{W}_2^2(\boldsymbol{p}_2, \boldsymbol{q}), \qquad (1)$$

where $\alpha \in [0, 1]$ and $\mathcal{W}_2(\boldsymbol{p}, \boldsymbol{q})$ denotes the ℓ_2 Euclidean Wasserstein distance between $\boldsymbol{p}$ and $\boldsymbol{q}$ (see Section 3). To obtain a sequence that morphs the distribution $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$ smoothly, a common approach is to solve Equation (1) for a linear series of α values, e.g. $\alpha \in \frac{1}{N+1}\{1, 2, ..., N\}$.

[1] Source code available at http://tiny.cc/barycenter



(b) Wasserstein barycenters

(a) Image #1

(c) GAN latent space linear interpolation

(e) Image #2

(d) Ours – WBP with a GAN as an image prior

Figure 2: Morphing a sports shoe to a boot using 3 methods, where $\alpha \in \{0, \frac{1}{8}, \frac{2}{8}, ..., 1\}$. In Figure 2b the intermediate images look blurry and unrealistic. In Figure 2c at first the shoe hardly changes and then immediately changes to a boot. Figure 2d demonstrates a smooth transition in both color and shape. Images taken from the Zappos50k dataset [2].

Indeed, solving the WBP for two input images, leads to a smooth (regular) and direct transition while avoiding ghosting artifacts.[2] That said, the intermediate samples do not necessarily seem "natural" as can be seen in Figures 1a and 2b. To overcome this issue, one may replace the ℓ_2 Euclidean metric with the geodesic distance over the manifold of natural images. However, this manifold is typically unknown or very complex, making this approach impractical.

In order to obtain natural intermediate images, recent works have suggested the use of Generative Adversarial Networks (GAN) [6, 7, 8, 9]. In this architecture, a generative network $G(\cdot)$ maps vectors $\boldsymbol{z}_i \in \mathbb{R}^m$, $m < n$ from a low dimensional latent space to high dimensional images. When two images and their matching latent representations are given, i.e. $G(\boldsymbol{z}_i) = \boldsymbol{x}_i$, a transition is obtained by linearly interpolating the two latent vectors as follows:

$$\boldsymbol{y}_i = G\left((1-\alpha_i)\boldsymbol{z}_1 + \alpha_i \boldsymbol{z}_2\right), \qquad (2)$$

with $\alpha_i = \frac{i}{N+1}$. Since each interpolated image $\boldsymbol{y}_i$ is an output of the generative network, each image follows an image prior, leading to a natural-looking transition. However, as we show in this work, these transitions do not necessarily obey the desired properties mentioned earlier. First, the pace of the changes might vary throughout the transformation as demonstrated in Figures 1b and 2c, where most of the transition is concentrated in one or two frames. Second, the change itself might not be direct and minimal. For example, in Figure 2c the colors become too bright before darkening back again.

The main contribution of this work is in providing a novel algorithm for solving a constrained form of the WBP. Furthermore, in this work we introduce a novel approach for image morphing that is based on the Euclidean WBP but with additional constraints on the obtained intermediate images. Concretely, we enforce each of the images in the sequence to reside on the manifold of natural images by using image priors, leading to a transition that fulfills all of the aforementioned requirements. Moreover, we present an approach to measure these three properties numerically and show the advantage of our method.

2. Previous Work

Image morphing has been studied and evolved for over three decades. Classical methods [10] have relied on a simple cross-dissolve operation together with a geometric warp of two images, using a dense correspondence map, which is typically hard to obtain automatically. In some cases, however, manual correspondence maps were avoided. For example, in [11], a method that is based on optimal-transport was suggested to obtain a short time domain interpolation, e.g. interpolating two consecutive video-frames. A recent work has suggested a morphing process in which the intermediate images are generated by patches of the input ones, constraining their similarity [12]. This method does not require such maps even when the input images differ substantially. Later, [9] has extended this concept by generalizing the local patch-based constraints to a single global one. To morph one image into the other, the authors have suggested to traverse the manifold of natural images. To this end, they first project the input images onto the latent space of a trained GAN, and then linearly interpolate these latent vectors. This transformation is used to compute a motion and color flow, which is then applied on one of the

[2] This usually requires a pre-processing normalization step.

inputs. Hence, the final transformation is actually a geometrical warp of the input image, as opposed to being generated by the model. In this work, we further extend this approach by traversing the latent space in a non-linear manner. This path is obtained by solving the Wasserstein barycenter problem for the input images. Moreover, we discard the use of the flow fields by increasing the resolution of the generated images. Furthermore, our method is not restricted to GANs and can be used with any image prior.

3. The Wasserstein Barycenter Problem

Before describing the WBP, we first provide a brief overview on optimal transport and Wasserstein distances. For an in-depth review of the topic, the reader is referred to [13] and [14].

3.1. Symbols and Notations

We define $\mathcal{D}$ as a space created by regular samples in $\mathbb{R}^2$, leading to an $n_1 \times n_2$ grid of pixels, where $n_1 n_2 = n$. For simplicity, we refer to $\boldsymbol{p} \in \mathcal{D}$ as a 1-dimensional vector of size n. The symbol $\mathcal{D}_+^1$ denotes the space of probability measures defined on $\mathcal{D}$, i.e. if $\boldsymbol{p} \in \mathcal{D}_+^1$, then $\sum_{i=1}^n p_i = 1$ and $p_i \geq 0$ where the element p_i is mapped to the i-th pixel in $\mathcal{D}$. Finally, we use $d_{\mathcal{D}}(i,j)$ to denote the Euclidean distance between pixels i and j in the grid defined by $\mathcal{D}$.

3.2. Optimal Transport

Given a source and target distributions $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{D}_+^1$, it is possible to transform one to the other using a transportation-plan $\boldsymbol{P} \in \mathbb{R}_+^{n \times n}$. This transportation plan describes the amount of mass to be passed from each pixel in $\boldsymbol{p}_1$ to each pixel in $\boldsymbol{p}_2$, while preserving mass conservation rules. The set containing all possible plans $U(\boldsymbol{p}_1, \boldsymbol{p}_1)$ is defined as:

$$U(\boldsymbol{p}_1, \boldsymbol{p}_1) = \left\{ \boldsymbol{P} \in \mathbb{R}_+^{n \times n} \middle| \boldsymbol{P} \cdot \mathbf{1}_n = \boldsymbol{p}_1 \cap \boldsymbol{P}^T \cdot \mathbf{1}_n = \boldsymbol{p}_2 \right\}, \tag{3}$$

where $\mathbf{1}_n$ is an all-ones vector of size n. For a given cost matrix $\boldsymbol{C} \in \mathbb{R}^{n \times n}$, *optimal transport* is defined as the transportation plan $\boldsymbol{P}^*$ which is the minimizer of:

$$\mathcal{L}_C(\boldsymbol{p}_1, \boldsymbol{p}_2) = \min_{\boldsymbol{P} \in U(\boldsymbol{p}_1, \boldsymbol{p}_2)} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{P}_{i,j} \boldsymbol{C}_{i,j}. \tag{4}$$

Specifically, when the matrix $\boldsymbol{C}$ is a distance matrix, then $\mathcal{L}_C$ is referred to as a *Wasserstein distance*. For example, when the Euclidean distance is used, Eq. (4) is simply:

$$\mathcal{L}_{d_{\mathcal{D}}}(\boldsymbol{p}_1, \boldsymbol{p}_2) = \min_{\boldsymbol{P} \in U(\boldsymbol{p}_1, \boldsymbol{p}_2)} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{P}_{i,j} d_{\mathcal{D}}(i,j), \tag{5}$$

and we denote $\mathcal{W}_2(\boldsymbol{p}_1, \boldsymbol{p}_2) \triangleq \sqrt{\mathcal{L}_{d_{\mathcal{D}}}(\boldsymbol{p}_1, \boldsymbol{p}_2)}$. Indeed, as the name suggests, the Wasserstein distance is a distance metric.

To find the minimizer $\boldsymbol{P}^*$ of Eq. (5), one needs to solve a LP problem of size $n \times n$. For example, an image of size 128×128 leads to a LP of size $16,384^2 \approx 2 \times 10^8$, making it an impractical task. To overcome this issue, we seek to approximate problem (5). A common approximation is the use of an entropic regularization [15]:

$$\mathcal{W}_2^2(\boldsymbol{p}_1, \boldsymbol{p}_2) = \min_{\boldsymbol{P} \in U(\boldsymbol{p}_1, \boldsymbol{p}_2)} \sum_{i=1}^n \sum_{j=1}^n \boldsymbol{P}_{i,j} d_{\mathcal{D}}(i,j) - \epsilon \mathbf{H}(\boldsymbol{P}), \tag{6}$$

$$\mathbf{H}(\boldsymbol{P}) \triangleq - \sum_{i,j} \boldsymbol{P}_{i,j} \left(\log \left(\boldsymbol{P}_{i,j} \right) - 1 \right). \tag{7}$$

This regularization stabilizes the solution by making the problem strictly convex and the solution can be found efficiently using the Sinkhorn algorithm [16]. Hereinafter, we denote $\mathcal{W}_2$ as the entropic-regularized Wasserstein distance.

3.3. Wasserstein Barycenters

For any given distance metric d, the *barycenter* of a set of inputs $\{\boldsymbol{x}_i\}_{i=1}^n$ and corresponding weights $\{w_i\}_{i=1}^n$ where $w_i \geq 0$ and $\sum_i w_i = 1$ is defined as:

$$\boldsymbol{x}_{\text{barycenter}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^n w_i d(\boldsymbol{x}, \boldsymbol{x}_i)^p, \tag{8}$$

where $p \geq 1$. Specifically, the Wasserstein barycenter problem is defined as the probability measure that minimizes the sum of p-powered Wasserstein distances to a set of probability measures $\{\boldsymbol{p}_i\}_{i=1}^n$:

$$\boldsymbol{q}_{\text{barycenter}} = \arg\min_{\boldsymbol{q} \in \mathcal{D}_+^1} \sum_{i=1}^n w_i \mathcal{W}_2^2(\boldsymbol{q}, \boldsymbol{p}_i), \tag{9}$$

where in (9) we chose $p = 2$. This problem is strictly convex and various efficient solvers have been suggested [3, 4, 17, 18]. Wasserstein barycenters have been used for various applications in image processing and shape analysis, including texture mixing [19], color transfer [20, 17] and shape interpolation [17]. In the following section, we propose a novel solution for a constrained version of the Wasserstein barycenter problem, and use it to obtain a natural-looking barycenter of images.

4. The Proposed Approach

To morph one image to the other, while obtaining natural looking intermediate images, we suggest to restrict the obtained images to satisfy some prior. Formally, we add a constraint to the barycenter problem (Eq. (1)) that limits the result to lie on a manifold $\mathcal{M}$:

$$\boldsymbol{p}_\alpha = \begin{cases} \arg\min_{\boldsymbol{q} \in \Sigma_n} \; (1-\alpha)\mathcal{W}_2^2(\boldsymbol{p}_1, \boldsymbol{q}) + \alpha \mathcal{W}_2^2(\boldsymbol{p}_2, \boldsymbol{q}) \\ \text{s.t.} \quad \boldsymbol{q} \in \mathcal{M}, \end{cases} \tag{10}$$

```
Algorithm 1: Constrained Wasserstein Barycenters
input      : Input densities $\{p_1, p_2\}$, an initial guess $\boldsymbol{q}^0$
             and a threshold $\epsilon$
output     : Constrained barycenter $\boldsymbol{q}$.
initialize: $\boldsymbol{u} \leftarrow \mathbf{0}$, $k \leftarrow 0$, $\boldsymbol{r}^0 \leftarrow \boldsymbol{q}^0$
repeat
    $\boldsymbol{q}^{k+1} \leftarrow \arg\min_{\boldsymbol{q}\in\Sigma_n} \alpha \mathcal{W}_2^2(\boldsymbol{p_1}, \boldsymbol{q}) + (1-\alpha)\mathcal{W}_2^2(\boldsymbol{p_2}, \boldsymbol{q}) + \frac{\mu}{2}\|\boldsymbol{r}^k - \boldsymbol{q} + \boldsymbol{u}^k\|_2^2$
    $\boldsymbol{r}^{k+1} \leftarrow \arg\min_{\boldsymbol{r}\in\mathcal{M}} \frac{\mu}{2}\|\boldsymbol{r} - \boldsymbol{q}^{k+1} + \boldsymbol{u}^k\|_2^2$
    $\boldsymbol{u}^{k+1} \leftarrow \boldsymbol{u}^k + \boldsymbol{r}^{k+1} - \boldsymbol{q}^{k+1}$
    $k \leftarrow k + 1$
until $\|\boldsymbol{q}^k - \boldsymbol{r}^k\| < \epsilon$;
return $\boldsymbol{q}^k$
```

As this problem might be hard to solve directly, we shall place an auxiliary variable $\boldsymbol{r} = \boldsymbol{q}$:

$$\boldsymbol{p}_\alpha = \begin{cases} \arg\min_{\boldsymbol{q}\in\Sigma_n, \boldsymbol{r}} \; (1-\alpha)\mathcal{W}_2^2(\boldsymbol{p}_1, \boldsymbol{q}) + \alpha\mathcal{W}_2^2(\boldsymbol{p}_2, \boldsymbol{q}) \\ \text{s.t.} \quad \boldsymbol{r} \in \mathcal{M}, \; \boldsymbol{r} = \boldsymbol{q}. \end{cases} \tag{11}$$

The Augmented Lagrangian of this problem is

$$\boldsymbol{p}_\alpha = \begin{cases} \arg\min_{\boldsymbol{q}\in\Sigma_n, \boldsymbol{r}, \boldsymbol{u}} \; (1-\alpha)\mathcal{W}_2^2(\boldsymbol{p}_1, \boldsymbol{q}) + \alpha\mathcal{W}_2^2(\boldsymbol{p}_2, \boldsymbol{q}) \\ \qquad\qquad + \frac{\mu}{2}\|\boldsymbol{r} - \boldsymbol{q} + \boldsymbol{u}\|_2^2 \\ \text{s.t.} \quad \boldsymbol{r} \in \mathcal{M}, \end{cases} \tag{12}$$

where $\mu > 0$. This optimization problem can be solved using the Alternating Direction Method of Multipliers (ADMM) [21], leading to the following steps (see Algorithm 1). First, we find a solution $\boldsymbol{q}$ to a regularized version of the WBP. This problem is strictly convex and has been previously studied. In our work we follow [4] which proposes a descent algorithm on the dual problem. The second step is a projection of the previous step result $\boldsymbol{q}$ onto the manifold $\mathcal{M}$. The third and the final step is a simple update of the dual variable $\boldsymbol{u}$. These steps are repeated until convergence is achieved. Figure 3 illustrates the differences between our approach and other image morphing approaches, specifically, when using a GAN as an image prior.

In cases where the manifold $\mathcal{M}$ is convex, the optimization problem (12) is convex, and convergence to a global minimum is guaranteed. That said, manifolds of interest, such as those of natural images, are often not convex (otherwise a simple linear interpolation between images would suffice) and therefore, only a local minimum is not guaranteed. Nevertheless, as we show in section 6, the obtained results are visually appealing. Note that this approach can be applied for a variety of priors, affecting only the second step in Algorithm 1. In the following subsections we demonstrate our method on the sparse prior and on GANs.

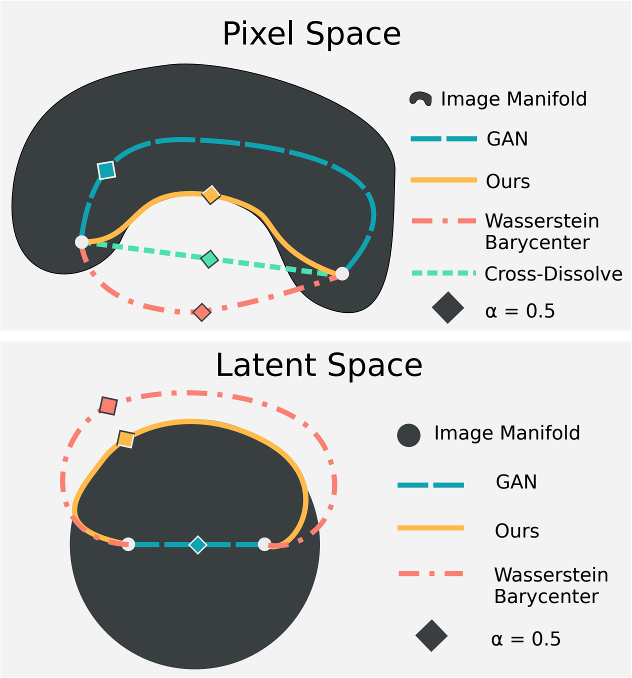


Figure 3: An illustration of the morphing process of an image, setting $\alpha = 0 \to 1$, using several approaches in the pixel space and the latent space of a trained GAN.

4.1. Sparse Prior

A well-known prior for various signal processing tasks is the sparse representation prior [22, 23, 24]. This model assumes that a signal $\boldsymbol{x} \in \mathbb{R}^n$ is constructed by a linear combination of only a few columns, also referred to as *atoms*, taken from a fixed matrix $\boldsymbol{D} \in \mathbb{R}^{n\times m}$, known as a *dictionary*. When a signal $\boldsymbol{y}$ is given, projecting it onto the model consists of finding its sparse representation vector $\boldsymbol{\alpha}$:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \text{ s.t. } \|\boldsymbol{\alpha}\|_0 < k, \tag{13}$$

for some $k \in \mathbb{N}$, typically much smaller than m. Generally, Eq. (13) is NP-hard [25] and various approximation algorithms have been suggested to solve it, such as the Orthogonal Matching Pursuit (OMP) and the Basis Pursuit (BP) algorithms [26, 27]. Once the representation vector $\hat{\boldsymbol{\alpha}}$ is found, the reconstructed signal is simply $\boldsymbol{y}^{\text{proj}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$.

In our approach, constraining the resulting barycenter image to satisfy the sparse representation prior, changes the second step in Algorithm 1 to a sparse coding algorithm, e.g. OMP. In our experiments, we further improve the visual results by approximating the MMSE estimator of $\hat{\boldsymbol{\alpha}}$ using stochastic resonance [28].

4.2. Generative Adversarial Networks

In the GAN setting [6, 7], a generative network $G(\boldsymbol{z}) : \mathbb{R}^m \to \mathbb{R}^n$ and a discriminative one $D(\boldsymbol{y}) : \mathbb{R}^n \to \{0, 1\}$ contest against each other. Given a dataset, the former is trained to generate samples from it when given a random input vector $\boldsymbol{z}$, while the latter is trained to distinguish the generated data from the original one. This approach leads to a model that is able to generate new data samples with statistical properties that are similar to the training set, by sampling random vectors $\boldsymbol{z}$ from the latent space of the model, and passing them through G.

In order to use the generative network for image morphing, an inverse mapping, i.e. a mapping from an input image $\boldsymbol{x}$ to its latent representation vector $\boldsymbol{z}$, is required. To obtain this mapping, we follow the approach described in [9] that we now briefly describe here. Once the generative network is trained, we train an encoder $E(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m$, such that $G(E(\boldsymbol{x}))$ is similar to the input $\boldsymbol{x}$:

$$E(\boldsymbol{x}) = \boldsymbol{z}^* = \min_{\boldsymbol{z}} \mathcal{L}\left(\boldsymbol{x}, G(\boldsymbol{z}(\boldsymbol{x}))\right), \tag{14}$$

where $\mathcal{L}$ is a pixel-wise ℓ_2 loss in simple cases such as MNIST [29]. For more complex images such as Zappos50k [2], the loss $\mathcal{L}$ is extended to a weighted sum of pixel-wise ℓ_2 and features extracted from AlexNet [30] trained on ImageNet [31]. This encoder-decoder scheme $G \circ E : \mathbb{R}^n \to \mathbb{R}^n$ may be perceived as a projection of the input signal onto the manifold of natural images. Therefore, To use GAN as a prior in our approach, the second step in Algorithm 1 is implemented by a simple feedforward activation of the obtained encoder-decoder.

5. Quantifying the Desired Properties

Above, we described three desired attributes for a natural looking image transformation: (i) to be smooth (regular), i.e. change at a constant pace; (ii) to be as minimal and direct as possible, avoiding unnecessary changes; and (iii) to include natural-looking images. To quantitatively show the advantage of our approach over other alternatives, we propose to measure each of these attributes as follows:

1. Regularity: to evaluate the smoothness of a transition, we propose to measure the distance between every two consecutive frames, and then compute the standard deviation of these distances over the entire transition. A steady paced transition results in a very low standard deviation, whereas irregular changes in the transformation correspond to a high variance. Since the Euclidean norm does not measure well movements of pixels [32], we adopt the Wasserstein distance for this task as it evaluates the minimal effort required to transport each pixel from one frame to the other.

2. Minimal: a minimal transition consists of a small number of pixel movements in the transformation process. As before, we adopt the Wasserstein distance for this purpose as it is a natural metric to quantify these movements. To evaluate the cost of the entire transition, we propose to average these distances between every two successive frames.

3. Natural looking images: to evaluate the affinity of an image to the class of natural images, we first train an autoencoder on a training set drawn from the chosen dataset. Once this model is trained, we feed each of the images generated in the transformation through the model, and compute their ℓ_2 distance to their own projection on the manifold characterized by the autoencoder.

6. Experiments

6.1. MNIST

We first demonstrate our method using the sparse representation model. From our experiments, the generative capabilities of this model seem inferior to those of newer alternatives such as GANs. Nevertheless, this example exposes the generality of our approach regarding the chosen prior. We start by learning a dictionary for the training set of the MNIST dataset [29], using online dictionary learning [33]. Then, we randomly select two test images of the same digit and morph one to the other using Algorithm 1, as described in Section 4.1. For comparison, we show the results of the morphing process using the unconstrained Wasserstein barycenters. As demonstrated in Figure 4, constraining the barycenter outcomes to satisfy the sparse prior yields sharper images that look like real digits, whereas the Wasserstein barycenter approach has no such guarantee.

We continue our experiments with employing GAN as an image prior, as described in Section 4.2. Specifically, we use the DCGAN architecture [7]. This prior is much more potent and is able to generate digit-like images, even when transforming between different digits. In this case, we experiment with barycenters of 4 input images. To do so, we modify Eq. (1) to include a convex combination of 4 Wasserstein distances, one from each source image. Figure 5 presents a comparison between our approach, the standard Wasserstein barycenters, and a bilinear interpolation of the latent vectors in the GAN setting. In contrast to the Wasserstein barycenters results, both our method and latent space bilinear interpolation produce natural digits. However, in the GAN setting, the pace is not consistent, leading to false barycenters. For example, in the first row, the image in the

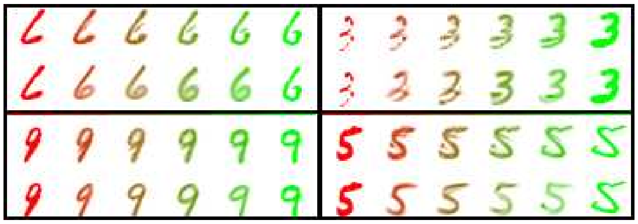


Figure 4: Wasserstein barycenters and our approach using a sparse prior in rows 1 and 2 of every subfigure respectively.

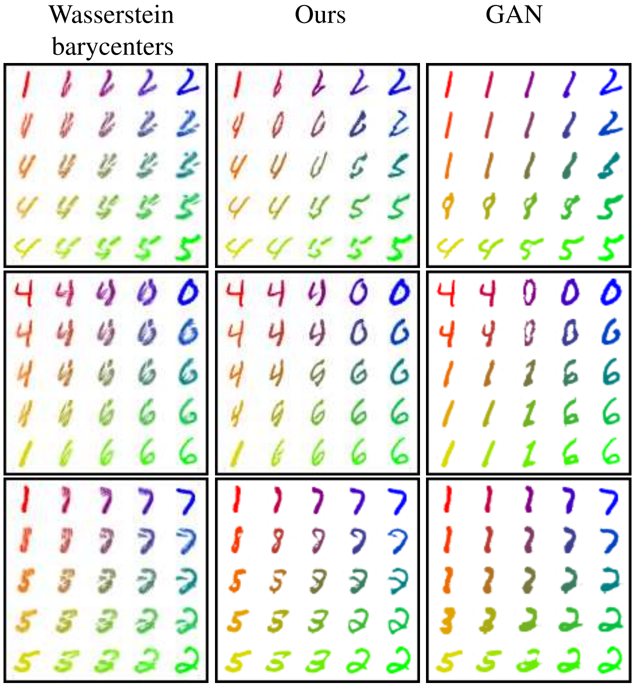


Figure 5: Barycenters of 4 input images, where each one is placed in a corner of the square. We compare 3 methods: Wasserstein barycenters, Algorithm 1 using DCGAN as an image prior, and DCGAN latent space bilinear interpolation. Colors are used to emphasize the interpolation.

center does not look like an "average" of all the others, but is rather similar to the digit "1", placed at the top-left corner.

6.2. Extended MNIST

The Extended-MNIST dataset [1] contains English characters that are more complicated than digits, and using the sparse prior leads to unpleasant results. Therefore, to obtain natural looking images, we focus our experiment in the DCGAN setting. Figures 1 and 6 demonstrate transitions using Wasserstein barycenters, linear interpolation in the latent space, and our method employing the DCGAN as the image prior. As before, it can be observed that the morphs obtained by the latent space interpolation do not result in a steady paced transition. At the bottom right example, the 'L' character hardly changes over the entire morphing process and most of the transformation occurs at the last 2 steps, i.e. $\alpha \in [0.8, 1]$. Furthermore, in the second example from the top on the left-hand side, it seems that transition from an 'r' to a 'J' in the latent interpolation case is not as straightforward as in our approach. Regarding the Wasserstein barycenter, the outcome images are blurry and often do not look like real English characters.

Method	Regularity	Total Dist.	Dist. to Manifold
Wasserstein Barycenters	0.033	10.95	4.96×10^{-4}
DCGAN	0.983	17.48	2.35×10^{-4}
Ours	0.232	12.58	2.37×10^{-4}

Table 1: The averaged quantified properties of 1500 randomly chosen transformations from the EMNIST dataset. In all parameters lower is better.

In addition to the provided visual results, Table 1 presents the averaged evaluation of all three methods, using the metrics specified in Section 5, on 1500 randomly chosen image pairs. These results show the advantage of our method as it obtained a straightforward and steadier paced transition compared to a latent space linear interpolation, while still being close to the desired manifold, as opposed to the Wasserstein barycenters approach.

6.3. Shoe Images – UT Zappos50K

The Zappos50K [2] is a much more complicated dataset compared to the previous two. Specifically, it contains more details and higher resolution. To train a GAN capable of generating such images, we split the training process in two, somewhat similar to the training scheme described in StackGAN [34]. First, we downscale the images to 64 × 64, and train a DCGAN model, as well as an encoder, as described in Section 4.2. The output images of this model are very smooth and lack fine high-frequency details. To add these details, we train an additional generative model. To this end, we generate a dataset of input-output image pairs as follows: the input images are the output of the encoder-decoder scheme, upsampled to 128 × 128, whereas the output images are the original ones downscaled to 128 × 128. This dataset is used as a training set to a pix2pix model [8]. To summarize, our projection scheme consists of the following stages: (i) project the input image to the DCGAN's latent space using a trained encoder; (ii) generate a low-frequency 64 × 64 image using a DCGAN; (iii) upscale the image to 128 × 128; and (iv) feed the image into a pix2pix model to generate high frequency details.

Once the models are trained, we compare the three following methods. The first is a standard Wasserstein barycenter solution (applied on each color channel sepa-

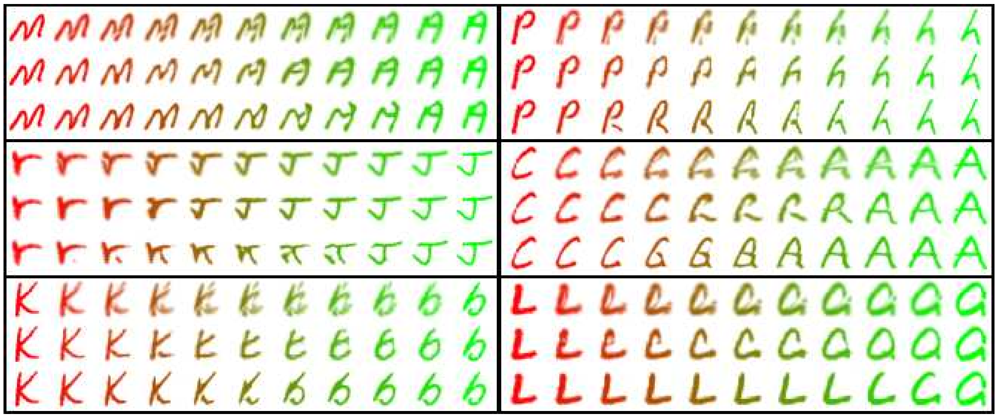


Figure 6: Morphing English character images using: Wasserstein barycenters, our approach using DCGAN as an image prior and DCGAN latent space linear interpolation in the 1-st, 2-nd and 3-rd row of every subfigure respectively.

rately). The second approach is our proposed algorithm, i.e. project each of the Wasserstein barycenters on the manifold of natural images, using the trained generative models, and the third is the common transition achieved using GANs as follows. Each input image is projected onto the latent space, using the encoder. Then, to obtain a transition, we linearly interpolate the two latent vectors and pass the interpolated vectors through the DCGAN and pix2pix models. From our experiments, iterating once produces the best results. The results of our experiments are presented in Figures 2 and 7. Both our method and the GAN alternative provide natural images most of the time. Furthermore, in cases where the two input images are similar in shape and color the difference between the two approaches seems mild (see the last example in Figure 7). However, when the contour or the hue of the two input images differ significantly, our approach brings on a much more steady and straightforward transition to both the shape and the colors of the images.

7. Discussion

In light of these results, we include a couple of remarks regarding the integration of our method in other GAN architectures and the complexity of our method before concluding this work.

Additional GAN architectures: integrating other models into Algorithm 1 only requires learning an encoder from the pixel space to the GAN's latent space. That said, it can be readily used with models that already posses an encoder-decoder structure, such as [35]. From our experiments, our approach is beneficial especially when the noise that generates the image has a simple distribution, as opposed to other architectures in which the noise is first transformed in some manner, such as in the case of StyleGAN [36].

Complexity: the second step in Algorithm 1 is usually quite fast, making the first step of the algorithm its main bottleneck. In our implementation, we chose to solve this optimization problem using an iterative algorithm [4], whose complexity grows with the number of pixels. Specifically, for n pixels, each iteration requires multiplying an $n \times n$ matrix with an n-sized vector. In our experiments, this approach led to a runtime of about 2-3 minutes to obtain a morphing process for images of size $128 \times 128 \times 3$ on an Intel i7-6900K CPU. Nevertheless, this complexity can be further improved. First, using convolutional Wasserstein distances [17], matrix multiplications can be replaced by a convolutional operator, omitting the need for storing the distances matrix and leading to a linear complexity in the number of pixels and a runtime of about one minute. Second, by migrating the implementation of this algorithm to a GPU, numerous barycenters can be computed in parallel [15]. Third, for input signals from a specific domain (such as images), the general algorithm [4] might be modified to improve performance. [37] is an example for this approach.

Conclusions: in this work we introduced a novel solution to a constrained variant of the well-known Wasserstein barycenter problem. While our algorithm is general, we propose to use it to obtain a natural barycenter of two or more input images, which can be used to generate a smooth transition from one to the other. For this purpose, we suggest constraining the barycenter to an image prior. Specifically, we demonstrate our approach using the sparse prior and GANs. We compare our method with the unconstrained variant of the WBP and a linear interpolation of latent vectors of GANs and show the advantage of the former in terms of the smoothness of the transition, the minimal quantity of changes, and the natural look of the acquired images both visually and numerically. We believe our approach of solving the WBP in its constrained form can be used in a variety of applications other than image morphing, e.g. pitch interpolation of two speakers, image style transfer and more, and we will focus our future work on such extensions.

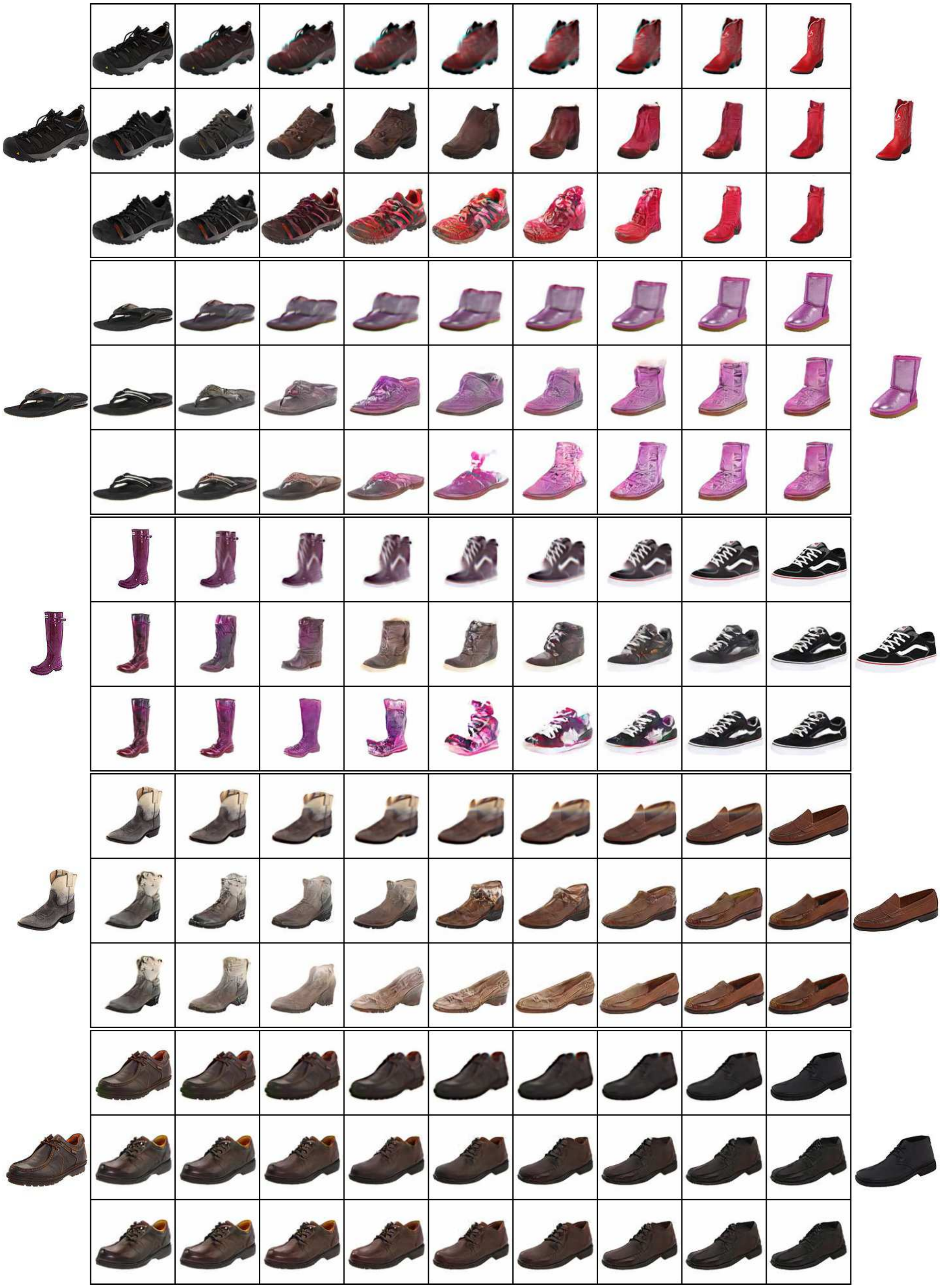


Figure 7: Morphing shoe images using: Wasserstein barycenters, our approach using GANs as an image prior and GAN latent space linear interpolation (1-st, 2-nd and 3-rd row of every subfigure respectively).

References

[1] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “Emnist: an extension of mnist to handwritten letters,” *arXiv preprint arXiv:1702.05373*, 2017. 1, 6

[2] A. Yu and K. Grauman, “Fine-grained visual comparisons with local learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 192–199, 2014. 2, 5, 6

[3] M. Cuturi and A. Doucet, “Fast computation of wasserstein barycenters,” in *International Conference on Machine Learning*, pp. 685–693, 2014. 1, 3

[4] M. Cuturi and G. Peyré, “A smoothed dual approach for variational wasserstein problems,” *SIAM Journal on Imaging Sciences*, vol. 9, no. 1, pp. 320–343, 2016. 1, 3, 4, 7

[5] L. Rüschendorf, “The wasserstein distance and approximation theorems,” *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985. 1

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014. 2, 5

[7] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *International Conference on Learning Representations*, 2016. 2, 5

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017. 2, 6

[9] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *European Conference on Computer Vision*, pp. 597–613, Springer, 2016. 2, 5

[10] G. Wolberg, “Image morphing: a survey,” *The visual computer*, vol. 14, no. 8, pp. 360–372, 1998. 2

[11] L. Zhu, Y. Yang, S. Haker, and A. Tannenbaum, “An image morphing technique based on optimal mass preserving mapping,” *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1481–1495, 2007. 2

[12] E. Shechtman, A. Rav-Acha, M. Irani, and S. Seitz, “Regenerative morphing,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 615–622, IEEE, 2010. 2

[13] G. Peyré, M. Cuturi, *et al.*, “Computational optimal transport,” *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019. 3

[14] C. Villani, *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008. 3

[15] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *Advances in Neural Information Processing Systems*, pp. 2292–2300, 2013. 3, 7

[16] R. Sinkhorn, “Diagonal equivalence to matrices with prescribed row and column sums,” *The American Mathematical Monthly*, vol. 74, no. 4, pp. 402–405, 1967. 3

[17] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, “Convolutional wasserstein distances: Efficient optimal transportation on geometric domains,” *ACM Transactions on Graphics*, vol. 34, no. 4, p. 66, 2015. 3, 7

[18] N. Bonneel, G. Peyré, and M. Cuturi, “Wasserstein barycentric coordinates: histogram regression using optimal transport.,” *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 71–1, 2016. 3

[19] J. Rabin, G. Peyré, J. Delon, and M. Bernot, “Wasserstein barycenter and its application to texture mixing,” in *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446, Springer, 2011. 3

[20] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, “Regularized discrete optimal transport,” *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014. 3

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011. 4

[22] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010. 4

[23] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009. 4

[24] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006. 4

[25] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995. 4

[26] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44, IEEE, 1993. 4

[27] S. Chen and D. Donoho, “Basis pursuit,” in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 41–44, IEEE, 1994. 4

[28] D. Simon, J. Sulam, Y. Romano, Y. M. Lu, and M. Elad, “Mmse approximation for sparse coding algorithms using stochastic resonance,” *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4597–4610, 2019. 5

[29] Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database,” *AT&T Labs*, vol. 2, p. 18, 2010. 5

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012. 5

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Ieee, 2009. 5

[32] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009. 5

[33] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010. 5

[34] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915, 2017. 6

[35] A. Heljakka, A. Solin, and J. Kannala, “Pioneer networks: Progressively growing generative autoencoder,” in *Asian Conference on Computer Vision*, pp. 22–38, Springer, 2018. 7

[36] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019. 7

[37] M. Bonafini and B. Schmitzer, “Domain decomposition for entropy regularized optimal transport,” *arXiv preprint arXiv:2001.10986*, 2020. 7