HAL open science

A Fast Multi-Layer Approximation to Semi-Discrete Optimal Transport

Arthur Leclaire, Julien Rabin

To cite this version:

Arthur Leclaire, Julien Rabin. A Fast Multi-Layer Approximation to Semi-Discrete Optimal Transport. Seventh International Conference on Scale Space and Variational Methods in Computer Vision, Jun 2019, Hofgeismar, Germany. hal-02080222

HAL Id: hal-02080222

https://hal.science/hal-02080222v1

Submitted on 26 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Fast Multi-Layer Approximation to Semi-Discrete Optimal Transport

Arthur Leclaire[1] and Julien Rabin[2]

[1] Institut de Mathématiques de Bordeaux, Université de Bordeaux
arthur.leclaire@math.u-bordeaux.fr, www.math.u-bordeaux.fr/~aleclaire
[2] Normandie Univ., ENSICAEN, CNRS, GREYC
julien.rabin@unicaen.fr, sites.google.com/site/rabinjulien

Abstract. The optimal transport (OT) framework has been largely used in inverse imaging and computer vision problems, as an interesting way to incorporate statistical constraints or priors. In recent years, OT has also been used in machine learning, mostly as a metric to compare probability distributions. This work addresses the semi-discrete OT problem where a continuous source distribution is matched to a discrete target distribution. We introduce a fast stochastic algorithm to approximate such a semi-discrete OT problem using a hierarchical multi-layer transport plan. This method allows for tractable computation in high-dimensional case and for large point-clouds, both during training and synthesis time. Experiments demonstrate its numerical advantage over multi-scale (or multi-level) methods. Applications to fast exemplar-based texture synthesis based on patch matching with two layers, also show stunning improvements over previous single layer approaches. This *shallow* model achieves comparable results with state-of-the-art deep learning methods, while being very compact, faster to train, and using a single image during training instead of a large dataset.

Keywords: Optimal Transport · Texture Synthesis · Patch matching

1 Introduction

Optimal transport (OT) [21,16] provides a powerful tool to measure the distance between two probability distributions. It has found several applications in image processing, for example feature matching [15] which requires comparing histograms of gradient orientations, color transfer [13] where OT distances are used to compare color distributions, texture mixing [14,22] which formulates as computing barycenters of texture models for the OT distance, shape interpolation [18] which uses OT distance on shapes identified on probability measures. Here we will focus on exemplar-based texture synthesis, which consists in producing a (possibly very large) image that has the same aspect as a given texture sample while not being a verbatim copy.

A common way to formulate texture synthesis is to ask for an image which is as random as possible while respecting a certain number of statistical constraints. Thus, texture synthesis can be addressed by exploiting OT distances to

compare distributions of linear or nonlinear filter responses [19], or comparing directly patch distributions [5,3]. In this paper, we build on the model proposed in [3] which performs texture synthesis by applying a well-chosen OT map on 3×3 patches in a coarse to fine manner (starting from a white noise at the lowest resolution). One limitation of this work is that the restriction to 3×3 patches prevents the proper reproduction of geometric structures at larger scales. Complementary to the multiresolution strategy, a common way of dealing with large geometric structures is to work with larger patches as is usually done in popular patch-based methods, as [8] (8×8 to 32×32 patches) or [2] (7×7 patches). But then we must face the fact that OT maps are very difficult to compute in such a high-dimensional setting.

Following [3] we are particularly interested in the semi-discrete OT case, where one wishes to send an absolutely continuous probability measure μ onto a discrete probability measure ν while minimizing a transportation cost. For the semi-discrete case, as shown in [1,7], the OT map can be obtained as a weighted nearest-neighbor (NN) assignment, which is parameterized by a finite-dimensional vector v that solves a concave optimization problem. Several first and second order schemes have been proposed to solve this problem, based on exact gradient computations [11,6,9,7]. Unfortunately, these exact methods are very difficult to apply in high dimension (or when the point set supporting ν is large) where exact gradients computations are intractable. To cope with that, one can use stochastic optimization techniques that rely on Monte-Carlo estimations of the gradient, as proposed in [4,3]. But such a stochastic method has a very slow convergence in high dimension, especially when the support of target measure has a large cardinal. One can improve convergence speed by working in a multi-scale framework, meaning that one precomputes a decomposition of the target measure *i.e.* several simplified versions of the target discrete distribution (with a clustering algorithm, for example K-means), and computes the OT map by progressively refining the target distribution. It has been noted in [11,9] that the approximate OT map obtained at one scale could be used to initialize the map at the next finer scale.

Here we propose an alternative way of exploiting this multi-scale framework. Indeed, we suggest to optimize a transport map having a parametric form which combines several layers of weighted NN assignments. To distinguish from previous works, such a map will be called a *multi-layer* transport map. Thanks to its inherent structure, during the iterative optimization algorithm, the multi-layer map attains a near-optimal transport cost in a much faster way than previous algorithm. But the price to pay is that we deviate from the truly optimal solution, meaning that the precomputed decomposition of the target measure introduces a bias that cannot be canceled (even at infinite time). As a byproduct, once estimated, such a multi-layer map can be applied much faster because it trades one NN projection on a large finite set by hierarchical NN projections on much smaller sets. Such properties are very profitable for imaging applications, where satisfying the marginal constraints is certainly more important than the true optimality in terms of transportation cost.

In Section 2 we first recall the framework of semi-discrete optimal transport. In Section 3 we detail the construction of multi-layer transport maps and their stochastic optimization. In Section 4, we propose numerical experiments that illustrate the benefits of these multi-layer maps to realize approximate OT: first, we illustrate on a one-dimensional example the gain in computational time to optimize the parameters; and next, we use these multi-layer maps to tackle difficult examples of OT in the space of large patches, which leads to remarkable results in texture synthesis.

2 Semi-discrete Optimal Transport

Given two probability measures μ, ν on $\mathbb{R}^D$, the Monge formulation of OT writes

$$\inf_T \int_{\mathbb{R}^D} \|x - T(x)\|^2 d\mu(x), \tag{OT-M}$$

where the infimum is taken over all measurable functions $T : \mathbb{R}^D \to \mathbb{R}^D$ for which the push-forward-measure $T_\sharp \mu$ equals ν. General conditions for existence and unicity of solutions can be found in [21] and [16]. In this paper we focus on the semi-discrete case, meaning that μ is an absolutely continuous probability distribution on $\mathbb{R}^D$ with density ρ and that ν is supported on $Y = \{y_j \ , \ j \in J\}$ finite with

$$\nu = \sum_{j \in J} \nu_j \delta_{y_j} \quad \text{s.t.} \quad \sum_{j \in J} \nu_j = 1 \text{ and } \nu_i \geq 0. \tag{1}$$

As proved in [1,9,7], the solution of (OT-M) is a weighted NN assignment

$$T_{Y,v}(x) := \operatorname*{argmin}_{y_j \in Y} \|x - y_j\|^2 - v(j). \tag{2}$$

This map is defined almost everywhere, and its preimages define a partition of $\mathbb{R}^D$ (up to a negligible set), called the *Laguerre cells*. The solution $T_{Y,v}$ is entirely parameterized by a vector v that maximizes the concave function $H_{Y,\nu}(v) := \mathbb{E}_{X \sim \mu}\left[h_{Y,\nu}(X, v)\right]$ where

$$h_{Y,\nu}(x, v) = \left(\min_{j \in J} \|x - y_j\|^2 - v(j)\right) + \sum_{j \in J} v(j)\nu_j, \tag{3}$$

so that we have for all $v \in \mathbb{R}^J$ and for almost all $x \in \mathbb{R}^D$,

$$\partial_{v(j)} h_{Y,\nu}(x, v) = -\mathbb{1}_{T_{Y,v}(x) = y_j} + \nu_j, \quad \text{and} \quad \partial_{v(j)} H_{Y,\nu}(v) = -\mu(T_{Y,v}^{-1}(y_j)) + \nu_j. \tag{4}$$

This important result has been used in [1,11,9,7] to optimize weights v when distributions μ and ν are defined on $\mathbb{R}^D$ for $D = 2$ or $D = 3$ dimensions, making use of gradient descent or L-BFGS algorithm. But in higher dimension, exact gradient computation is challenging, and thus following [4,3], one may turn to the Average Stochastic Gradient Descent (ASGD) Algorithm 1 to solve it.

Algorithm 1: ASGD to estimate OT map $T_{Y,v}$ that solves (OT-M)

Inputs: source density μ, target discrete distribution ν,
initial assignment weight (*e.g.* $\tilde{v} = 0$ and $v = 0$), and gradient step (*e.g.* $C = 1$)
for $t = 1, \dots, T$ **do**
  Draw a sample $x \sim \mu$
  Compute the gradient $g \leftarrow \nabla_v h_{Y,\nu}(x, \tilde{v})$ (see Eq. (4))
  Gradient ascent of weights: $\tilde{v} \leftarrow \tilde{v} + \frac{C}{\sqrt{t}} g$
  Average of updates: $v \leftarrow \frac{t-1}{t} v + \frac{1}{t} \tilde{v}$
return v

3 Multi-layer Approximation of the Transport Map

A classical approach in signal processing is to use multi-scale representation to accelerate or handle large data. This strategy has recently gained some interest to compute approximate or exact optimal transport maps in large discrete or semi-discrete problems [11,12,17,9] to overcome some convergence speed issues.

3.1 Multi-scale Semi-Discrete Optimal Transport

The discrete target measure ν can be decomposed from fine ($\ell = 0$) to coarse ($\ell = L - 1$) scales by setting

$$\nu^\ell = \sum_{j \in J^\ell} \nu_j^\ell \delta_{y_j^\ell}, \quad \text{supported by } Y^\ell = \{ \ y_j^\ell \ , \ j \in J^\ell \ \}. \tag{5}$$

Starting from $\nu^0 = \nu$ (and $Y^0 = Y$), the measure $\nu^{\ell+1}$ should be a close approximation of ν^ℓ with a decreasing budget support composed of $|J^{\ell+1}|$ points.

In practice, following [11], we first extract (for example with the K-means algorithm) a clustering $Y^\ell = \bigsqcup_{j \in J^{\ell+1}} C_j^\ell$ of the support Y^ℓ of ν^ℓ, and we denote by $Y^{\ell+1} = \{y_j^{\ell+1}, j \in J^{\ell+1}\}$ the corresponding cluster centroids. Next, we compute the weights by gathering the mass in each cluster $\nu_j^{\ell+1} = \nu^\ell(C_j^\ell)$. An illustration of this multi-scale clustering is given in Figure 1.

The authors of [11,9] propose to estimate the optimal transportation map $T_{Y,v}$ from μ towards ν at the finest scale ($\ell = 0$) using a bottom-up strategy (from $\ell = L - 1$ to $\ell = 0$) that sequentially estimates the maps T_{Y^ℓ, v^ℓ} that realizes the OT between μ and ν^ℓ. The idea is to accelerate the optimization of v^ℓ by starting from an initial estimate that is extrapolated from the solution $v^{\ell+1}$ computed at the previous scale. While this method computes an exact solution, some heuristics is needed to extrapolate v^ℓ from $v^{\ell+1}$ (for example in [11], propagating the value $v^{\ell+1}(j)$ associated to centroid $y_j^{\ell+1}$ to the points in the corresponding cluster C_j^ℓ).

Let us also briefly mention that multi-scale strategies have been proposed to deal with *discrete* OT problems (*i.e.* when both distributions are discrete)

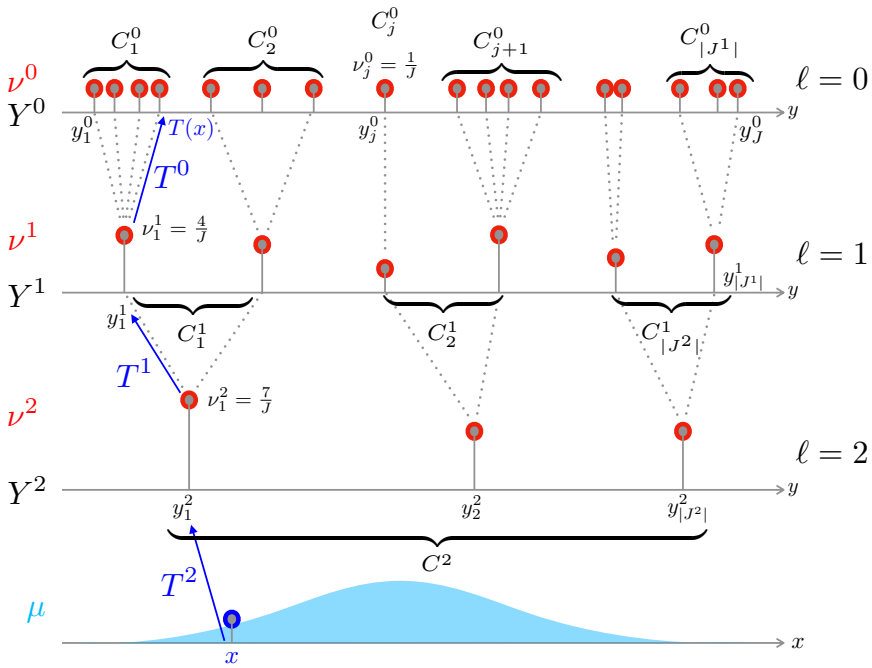


Fig. 1. Illustration of the multi-scale approximation of the discrete target distribution ν in the 1D case, and the multi-layer transport map $T(x)$ applied sequentially with $L = 3$ layers. See the text for more details about notation.

with very large data sets (yet in small dimension): an exact multiscale approach from [17], an approximation method in [12], and multi-level OT with Wasserstein-1 [10].

3.2 Multi-layer Transport Map

In this work, we consider a different approach that *approximates* the OT map with multi-layer transport maps to accelerate computation time for both the estimation of the mapping (off-line) and the mapping itself (on-line). The main difference with [11] is that the multi-scale representation of the target distribution (as illustrated in Figure 1) is used to sequentially estimate a *hierarchical* Laguerre cell partitioning of the source distribution. This hierarchical clustering induces a tree structure, and thus applying the multi-layer map consists in a tree search which is faster than a direct global NN projection. More precisely, the proposed approach is as follows:

– **Training:** We estimate simultaneously along the tree structure, a weight vector $v_{j^\ell}^\ell$ defining the transport $T_{C_{j^\ell}^\ell, v_{j^\ell}^\ell}$, which maps the corresponding Laguerre cell indexed by $j^{\ell+1}$ of the source distribution μ (defined itself at the

previous layer $\ell+1$ by a centroid $y^{\ell+1}_{j^{\ell+1}}$) to the cluster $C^{\ell}_{j^\ell}$ of points from the multi-scale target distribution ν^ℓ. The relationship between indices j^ℓ and $j^{\ell+1}$ depends entirely on the fixed hierarchical clustering that is performed beforehand, and is detailed in the paragraph hereafter.

– **Synthesis:** To find the image $T(x) \in Y$ of a point x, one should trace back the hierarchy of Laguerre cells to which x belongs and then apply the finest estimated transport map.

Eventually, the multi-layer map takes the form

$$T^\ell(x) = T_{C^\ell_{j^\ell}, v^\ell_{j^\ell}}(x) \tag{6}$$

where j^ℓ indicates the index of the cluster the point x is assigned to at layer ℓ. These indices are *recursively* defined (starting from $\ell = L-1$ where there is only one root cluster $C^{L-1}_{\cdot} = Y^{L-1}$, as illustrated in Fig. 1), namely $j^\ell(x)$ (denoted simply j^ℓ for the sake of brevity) is computed from the weighted NN for the previous layer $\ell+1$

$$j^\ell = \underset{j \mid y^{\ell+1}_j \in C^{\ell+1}_{j^{\ell+1}}}{\operatorname{argmin}} \|x - y^{\ell+1}_j\|^2 - v^{\ell+1}_{j^{\ell+1}}(j) \ . \tag{7}$$

Notice that the final value of the multi-layer map is $T(x) = T^0(x)$, but its evaluation requires to compute L transport maps $T^\ell(x)$ (actually the sequence of indices $j^{\ell+1}(x)$) from $\ell = L-1$ to $\ell = 0$. In the experiments shown in this paper, we restrict to the case of only two layers, which is illustrated in Fig. 2.

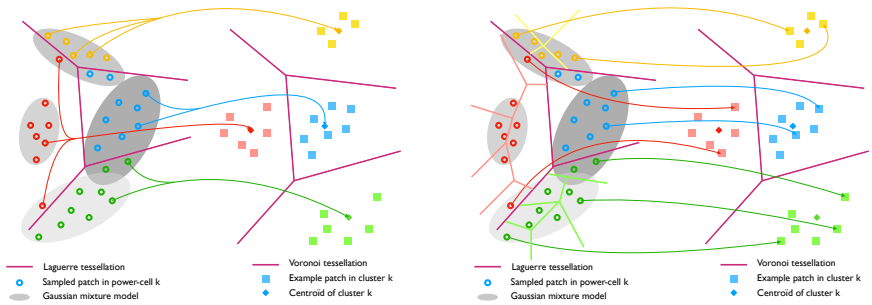
The weights $v^\ell_{j^\ell}$ (for all $\ell \in \{0, \dots, L-1\}$ and $j^\ell \in J^\ell$) are optimized so that it realizes the semi-discrete OT between the restriction of μ to the Laguerre cell $\mathcal{L}^\ell_{j^\ell} = (j^\ell)^{-1}(j^\ell(x))$ and the restriction $\nu^\ell_{j^\ell}$ of ν^ℓ to $C^\ell_{j^\ell}$. In other words, assuming that $T^{\ell+1}$ is fixed (*i.e.* for a given $j^{\ell+1}$), we seek to maximize

$$H^\ell(v^\ell) = \sum_{j^\ell \in J^\ell} \int_{\mathcal{L}^\ell_{j^\ell}} \min_{\substack{j \\ y^\ell_j \in C^\ell_{j^\ell}}} \left(\|x - y^\ell_j\|^2 - v^\ell_{j^\ell}(j) \right) d\mu(x) + \sum_{\substack{j \\ y^\ell_j \in C^\ell_{j^\ell}}} v^\ell_{j^\ell}(j) \nu^\ell_j. \tag{8}$$

Notice that this cost is composed of $|J^{\ell+1}|$ separate semi-discrete OT problems with restrictions of μ and ν^ℓ. Indeed, the problem (8) is the dual convex of the Kantorovich formulation of OT between those restricted measures. Writing instead the Monge formulation of these separates sub-problems, and denoting by S^ℓ the map that gathers the restrictions $T_{C^\ell_{j^\ell}, v^\ell_{j^\ell}}$ on the cells $\mathcal{L}^\ell_{j^\ell}$, we get that S^ℓ minimizes the usual cost

$$\int_{\mathbb{R}^D} \|S^\ell(x) - x\|^2 d\mu(x) = \sum_{j^\ell \in J^\ell} \int_{\mathcal{L}^\ell_{j^\ell}} \|S^\ell(x) - x\|^2 d\mu(x) \tag{9}$$

but with marginal constraints on each of the Laguerre cells $\mathcal{L}^\ell_{j^\ell}$. We can apply ASGD to treat each of these sub-problems. It requires to sample the restricted



First layer $\ell = 1$: the coarse transport map T^1 maps every points of a Laguerre cell j of μ to the centroid of the corresponding cluster C_j^0 in ν.

Second layer $\ell = 0$: the fine mapping $T = T^0$ is defined from the transport maps $T_{C_j^0, v_j^\ell}$.

Fig. 2. Illustration of a multi-layer map (for $L = 2$ layers). Here the source distribution μ is chosen to be a Gaussian mixture model with 4 components (in graylevels). For each layer ℓ, the arrows illustrate $T^\ell(x)$ (arrows) the multi-layer mapping of samples x drawn from μ (circle points on the left) to the points of the discrete distribution ν^ℓ (diamonds for layer $\ell = 1$ and square for layer $\ell = 0$).

measure μ. But in practice, we treat simultaneously all the sub-problems in all layers: at each new sample we update only the gradient relative to the current active Laguerre cell of the hierarchy, which in the end is equivalent to sample all restricted measures. This optimization procedure is summarized in Algorithm 2.

Algorithm 2: ASGD for the estimation of the multi-layer map $T_{Y,v}$.

Inputs: source density μ, target distribution ν, gradient step C, number of layers L and number of iterations T

Hierarchical clustering $\{\nu^\ell, \ell = 0, \dots, L-1\}$ of ν
Set $v_j^\ell \leftarrow 0$, $\forall\, \ell, j$ (weights initialization)
Set $n_j^\ell \leftarrow 0$, $\forall\, \ell, j$ (number of visits in cluster C_j^ℓ)
for $t = 1, \dots, T$ **do**
  Draw a sample $x \sim \mu$
  for $\ell = L-1, \dots, 0$ **do**
    Compute the corresponding cluster index: $j \leftarrow j^\ell(x)$
    $n_j^\ell \leftarrow n_j^\ell + 1$
    $g \leftarrow \nabla_v h_{C_j^\ell, \nu_j^\ell}(x, \tilde{v}_j^\ell)$ (in Eq. (4))
    $\tilde{v}_j^\ell \leftarrow \tilde{v}_j^\ell + \frac{C}{\sqrt{n_j^\ell}} g$
    $v_j^\ell \leftarrow v_j^\ell + \frac{1}{n_j^\ell}\left(\tilde{v}_j^\ell - v_j^\ell\right)$

Outputs: $\{\nu^\ell\}_{\ell \le L}$ and $\{v^\ell\}_{\ell \le L}$

4 Numerical Experiments

In this section we illustrate that multi-layer transport maps can be used to approximate semi-discrete optimal transport, and that they can be computed more efficiently. We start with a simple one-dimensional example (where the true solution of optimal transport is explicit) and next turn to some imaging applications in higher dimension.

4.1 One-dimensional Example

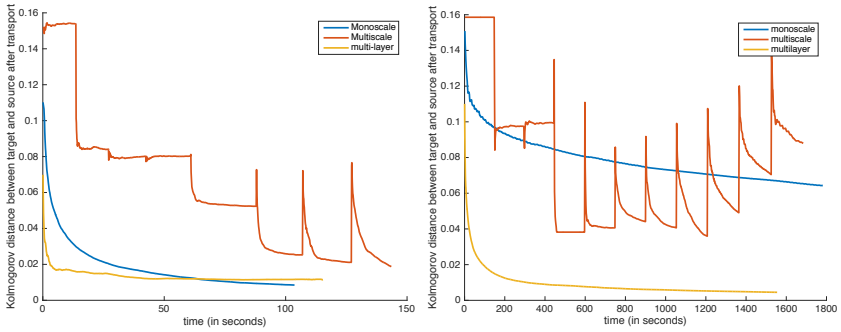
Here we consider the optimal transport problem when $D = 1$, between the normalized Gauss distribution $\mu \sim \mathcal{N}(0, 1)$ and the discrete uniform distribution ν on $J = 10^3$ (then $J = 10^4$) equally spaced points between -1 and 1 (see illustration in Figure 1). We will compare the simple ASGD method (Algorithm 1) to the proposed multi-layer scheme (Algorithm 2), and also to the multi-scale approach of [11]. For the multi-layer scheme, we only consider two layers. For that we use only one clustering of the target points, with 10 clusters (computed with the K-means algorithm). For the multi-scale scheme of [11], except for the finest scale, the transported measure is obtained by extrapolating the weights to the corresponding cluster.

The comparison will focus on the distance between the transported measure $T_\sharp\mu$ and the target measure ν (which does not entirely reflect the optimality in terms of transportation cost). We will use the Kolmogorov distance between those two discrete measures, that is, the L^∞ distance between the cumulative distribution functions (cdf). The computation of the cdf for the mono-layer map has already been explained in [3], and it can be extended to the multi-layer map. Notice that the cost of one iteration is not the same for those three iterative algorithms. Thus we will compare the evolution of the distance with respect to the true computation time (in seconds) and not the number of iterations (approximately 10^5 iterations for $J = 1000$, and 10^6 for $J = 10\,000$). The results are reported in Figure 3.

One can observe that the multi-layer scheme quickly realizes a good approximation of the target distribution, much faster than the mono-scale. This advantage is greater as the number of points increases, making this approach interesting to estimate and approximate transport for large set of points. However, one can observe that the limiting value for the multi-layer scheme is larger than the one with the mono-scale version, which reflects the bias that is introduced by imposing the multi-layer form of the transportation plan (associated to one fixed decomposition of ν). The multi-scale approach does not perform well in this scenario, where the error can increase when changing scale. One explanation may be that the initialization from the previous scale should be handled more carefully.

4.2 Texture Synthesis

To illustrate the interest of the proposed multi-layer approach, we use the multi-scale patch-based texture synthesis framework proposed in [3], where the semi-



$J = 1000$ points $J = 10\,000$ points

Fig. 3. We monitor the evolution of the distance between $T_\sharp\mu$ and ν during the three iterative algorithms explained in the text. The horizontal axis represents the computational time (in seconds) and the vertical axis the Kolmogorov distance between the current transported measure $T_\sharp\mu$ and the target measure ν. The number of clusters for the multi-layer approach is $|J^1| = \frac{J}{10}$. See the text for comments.

discrete OT is now estimated with Algorithm 2 instead of the single scale Algorithm 1. This enables to use larger patches (7×7 color patches instead of 3×3 in [3], so that dimension is $D = 147$), as well as a larger number of points (up to $J = 16\,000$ instead of $J = 1000$ in [3]). Additionally, as the patches are larger, we use Gaussian weighting for aggregation to avoid excessive blurring when averaging patch, and we also use a final optimal transportation at the finest resolution with 3×3 patches to restore fine details that are lost for the same reason. See Figure 4 for an illustration.

Parameters for both methods are fixed through experiments: $S = 4$ resolutions and a Gaussian mixture model with 10 components are used for the texture model (see [3] for details); the patch size is $w \times w$ ($w = 3$ for 1-layer OT and $w = 7$ for bi-layer OT); $J = 1000$ patches are randomly sampled from the example image when using 1-layer OT, and $J = 1000, 2000, 4000, 16\,000$ (resp.) patches (from coarse to fine resolution) when using bi-layer OT, using $|J^1| = 10, 10, 20, 40$ (resp.) clusters that are estimated by a K-means algorithm at the first layer); a centered Gaussian weight is used for patch averaging of patch with bi-level OT, with a standard deviation $\sigma = \frac{w}{2}$; Algorithm 1 and Algorithm 2 are used with 10^6 iterations.

Figure 5 illustrates the interest of the multi-layer patch transportation over the single-layer approach originally proposed in [3]. As already mentioned, the latter is restricted to small patch dimension due to convergence issues. The use of large patch makes it possible to capture and synthesize larger scale structures from the example image.

Figure 6 shows a small comparison with the state-of-the-art Texture Networks [20] approach. This method first trains a convolutional network generator using an example image and a deep neural network feature extractor (VGG-19) that is trained separately on a very large classification dataset of natural images

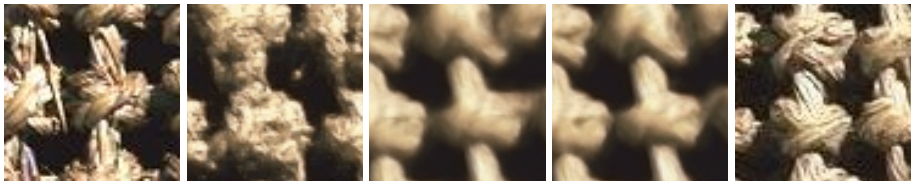


Fig. 4. Limitations of the 1-layer OT model of [3] for texture synthesis when using large patches. Images are cropped to display small details, full images are shown in Fig. 5. From left to right: example image, result from [3] (with 3×3 patches), result when using 7×7 patches without/with Gaussian weighting, and the proposed approach.

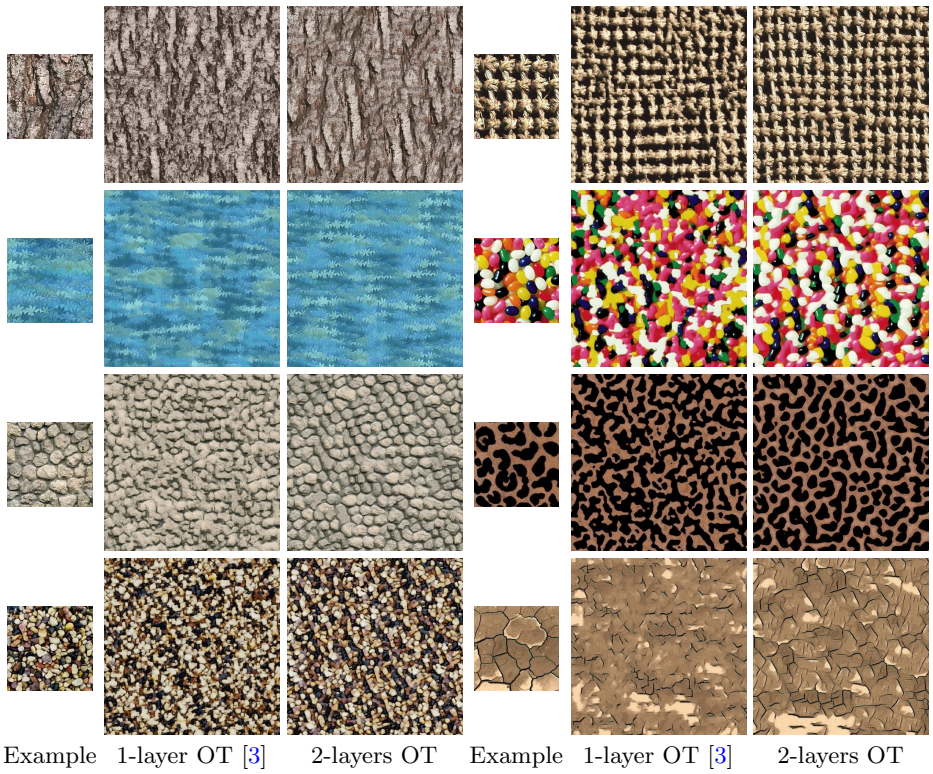
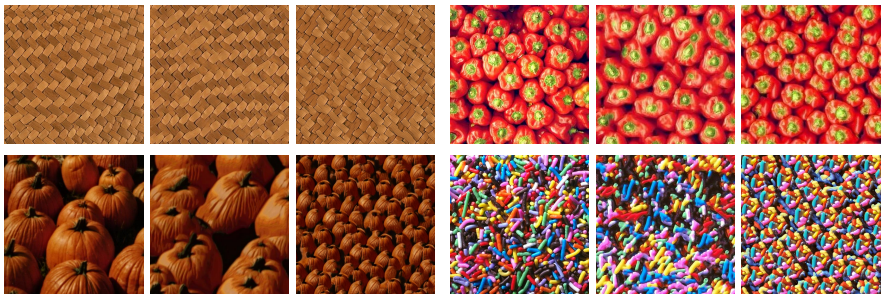


Fig. 5. Texture synthesis comparison between the 1-layer OT model of [3] and our 2-layers OT model.

(ImageNet), and then performs fast texture synthesis by feeding the generator with a random input. While both methods produce similar results on regular pattern (first row of Fig. 6), it is not the case for other type of textures (second row), where the method of [20] tends to generate pseudo-periodic images. Note that, while our approach does not requires the use of a GPU for the training

stage, both methods are completely parallel and achieve similar computation time during synthesis (within a second on a GPU for 1024×1024 images).



Examples 2-layer OT [20] Examples 2-layer OT [20]

Fig. 6. Texture synthesis comparison between the proposed approach (multi-layers Optimal Transport of 7×7 patches) and Texture Networks [20] (a feed-forward convolutional neural network). Note that for the *pumpkins* example, $S = 6$ resolutions has been used instead of $S = 4$ in the rest of experiments to capture large scale object information.

5 Discussion and conclusion

We have described a new strategy to approximate the semi-discrete optimal transport problem based on hierarchical multi-layer transport maps. A simple stochastic algorithm has been proposed and shown to be effective with only two layers. It allows for faster training and synthesis than previous multi-scale approaches, and makes it also possible to deal with larger point sets in higher dimension. Its application to texture synthesis demonstrates its practical interest, making it possible to compete with recent machine learning techniques (based on deep convolutional neural networks).

In this preliminary work, we have only experimented with bi-layer transport maps, but we expect the proposed model to be even more efficient when exploiting its full multi-layer potential, as done by deep learning techniques. However, as shown in experiments for the multi-scale approach, this raises the problem of the optimal setting of the hierarchical structure (similarly to the importance of network design in machine learning) to avoid strong bias in the approximation of the optimal transportation map. Besides, the convergence of the stochastic algorithm for estimating multi-layer transport maps remains to be investigated, since it does not correspond to a convex problem anymore.

Acknowledgments

This project has been carried out with support from the French State, managed by the French National Research Agency (ANR-16-CE33-0010-01).

References

1. Aurenhammer, F., Hoffmann, F., Aronov, B.: Minkowski-type theorems and least-squares clustering. Algorithmica **20**(1), 61–76 (1998)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics-TOG **28**(3), 24 (2009)
3. Galerne, B., Leclaire, A., Rabin, J.: A texture synthesis model based on semi-discrete optimal transport in patch space. SIAM Journal on Imaging Sciences **11**(4), 2456–2493 (2018)
4. Genevay, A., Cuturi, M., Peyré, G., Bach, F.: Stochastic optimization for large-scale optimal transport. In: Proc. of NIPS. pp. 3432–3440 (2016)
5. Gutierrez, J., Galerne, B., Rabin, J., Hurtut, T.: Optimal patch assignment for statistically constrained texture synthesis. In: Proceedings of SSVM (2017)
6. Kitagawa, J.: An iterative scheme for solving the optimal transportation problem. Calculus of Variations and Partial Differential Equations **51**(1-2), 243–263 (2014)
7. Kitagawa, J., Mérigot, Q., Thibert, B.: A Newton algorithm for semi-discrete optimal transport. Journal of the European Math Society (2017)
8. Kwatra, V., Essa, I., Bobick, A., Kwatra, N.: Texture optimization for example-based synthesis. ACM TOG **24**(3), 795–802 (Jul 2005)
9. Lévy, B.: A numerical algorithm for L2 semi-discrete optimal transport in 3D. ESAIM: M2AN **49**(6), 1693–1715 (2015)
10. Liu, J., Yin, W., Li, W., Chow, Y.T.: Multilevel optimal transport: a fast approximation of wasserstein-1 distances. arXiv preprint arXiv:1810.00118 (2018)
11. Mérigot, Q.: A multiscale approach to optimal transport. Computer Graphics Forum **30**(5), 1583–1592 (2011)
12. Oberman, A.M., Ruan, Y.: An efficient linear programming method for optimal transportation. arXiv preprint arXiv:1509.03668 (2015)
13. Rabin, J., Peyré, G.: Wasserstein regularization of imaging problems. In: ICIP 2011: 2011 IEEE International Conference on Image Processing (2011)
14. Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: Proceedings of SSVM. pp. 435–446 (2012)
15. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Computer Vision, 1998. Sixth International Conference on. pp. 59–66. IEEE (1998)
16. Santambrogio, F.: Optimal transport for applied mathematicians. Birkäuser, NY (2015)
17. Schmitzer, B.: A sparse multiscale algorithm for dense optimal transport. Journal of Mathematical Imaging and Vision **56**(2), 238–259 (2016)
18. Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. ACM Transactions on Graphics (TOG) **34**(4), 66 (2015)
19. Tartavel, G., Gousseau, Y., Peyré, G.: Variational texture synthesis with sparsity and spectrum constraints. Journal of Mathematical Imaging and Vision **52**(1), 124–144 (2015)
20. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.: Texture networks: feed-forward synthesis of textures and stylized images. In: Proc. of the Int. Conf. on Machine Learning. vol. 48, pp. 1349–1357 (2016)
21. Villani, C.: Topics in Optimal Transportation. American Math. Society (2003)
22. Xia, G., Ferradans, S., Peyré, G., Aujol, J.: Synthesizing and Mixing Stationary Gaussian Texture Models. SIAM J. on Imaging Sciences **7**(1), 476–508 (2014)