Wasserstein Barycenter and its Application to Texture Mixing

Rabin Julien, Gabriel Peyré, Julie Delon, Bernot Marc

Wasserstein Barycenter and its Application to Texture Mixing

Julien Rabin[1⋆], Gabriel Peyré[1], Julie Delon[2], and Marc Bernot[3]

[1] Ceremade, Univ. Paris-Dauphine
[2] LTCI, Telecom ParisTech
[3] Thales Alenia Space
{rabin,peyre}@ceremade.dauphine.fr delon@enst.fr
marc.bernot@thalesaleniaspace.com

Abstract. This paper proposes a new definition of the averaging of discrete probability distributions as a barycenter over the Wasserstein space. Replacing the Wasserstein original metric by a sliced approximation over 1D distributions allows us to use a fast stochastic gradient descent algorithm. This new notion of barycenter of probabilities is likely to find applications in computer vision where one wants to average features defined as distributions. We show an application to texture synthesis and mixing, where a texture is characterized by the distribution of the response to a multiscale oriented filter bank. This leads to a simple way to navigate over a convex domain of color textures.

1 Introduction

This paper considers the use of optimal transportation methods [1] in image synthesis. Optimal transportation has been extensively used as a distance to compare histogram features, see for instance [2].

Another interesting aspect of transportation approaches is the transportation mapping itself, which is investigated in this paper. Indeed, it allows various image modifications, such as *e.g.* color transfer [3], texture mapping [4], or contrast equalization of video [5] (see [6] for other applications).

1.1 Texture Synthesis

Texture synthesis is a popular problem in computer graphics, which consists in synthesizing a new image f visually similar to a given exemplar f^0.

Texture synthesis by recopy. Synthesis with high fidelity to the exemplar is performed by copying pixels with some coherence constraints on small patches [7, 8]. The quality of the synthesis is improved by copying patches or more general sets of pixels [9–13].

⋆ This work has been done with the support of the French "Agence Nationale de la Recherche" (ANR), under grant NatImages (ANR-08-EMER-009), "Adaptivity for natural images and texture representations".

Texture synthesis by statistical modeling. While copy-based methods probably yield the best synthesis quality, they often copy large blocks from the original input, and offer little or indirect control about the synthesis process.

Procedural methods use parametric models of textures, for instance built on top of a Gaussian noise [14]. They are popular in image synthesis because of there ease of use and low computational cost.

Texture modeling considers sets of statistical constraints learned from the exemplar, and use the stationarity of the texture for the estimation. Popular approaches use Markov random fields [15, 16] or Gibbs distributions built on top of multiscale filters [17].

The wavelet decomposition is often use to build statistical models, with first order histograms [18, 19] or higher order constraints [20].

Texture mixing. The texture mixing problem consists in synthesizing a new texture from a collection $\{f^j\}_{j\in J}$ of exemplars. The mixing should integrate in a meaningful way the colors and texture attributes of the exemplars.

Heeger and Bergen [18] and Bar-Joseph *et al.* [21] perform the mixing by combining multiscale wavelet coefficients. Averaging statistics of grouplet coefficients enables the mixing of geometrical turbulent textures [22].

Patch-based methods perform mixing using patches from the set of exemplars, which creates non-homogeneous textures, see for instance [23, 9, 13]. These methods tend to produces clusters of features and offer little understanding about the mixing process and how to control it.

Texture metamorphosis approaches [24–26] perform the mixing by finding correspondences between elementary features (or textons) between the textures, and progressively morphing between the shapes of the features. These methods are extended by Matusik et al. [27] to perform convex combination of textures, by warping patches and averaging 1D histograms.

1.2 Contributions

The main theoretical contributions of this work are the introduction of a novel definition for the barycenter of statistical distributions, together with an approximate definition more amenable for numerical computations. We introduce a stochastic gragient descent algorithm to compute the Wasserstein barycenter. The same algorithm is also used to compute the projection of an arbitrary distribution on the centroid. The last contribution of the paper is a general framework for the statistical synthesis of color textures, that encompasses several existing texture models as particular cases. This general framework, together with the Wasserstein barycenter computation, allows to perform color texture mixing.

2 Wasserstein Distance and its Approximation

This paper considers discrete density distributions in $\mathbb{R}^d$ that are represented as point clouds $X = \{X_i\}_{i\in I} \subset \mathbb{R}^d$. Since any permutation of X corresponds to the

same distribution, one considers metrics taking into account

$$[X] = \left\{ (X_{\sigma(j)})_{j=0}^{N-1} \setminus \sigma \in \Sigma_N \right\}, \tag{1}$$

where Σ_N is the set of all permutations of N elements. The methods developed in the paper can be extended to weighted point clouds and density defined continuously over $\mathbb{R}^d$.

2.1 Wasserstein Distance

The quadratic Wasserstein distance $W(X,Y)$ between two point clouds of same size $|I| = N$ is defined as

$$W(X,Y)^2 = \min_{\sigma \in \Sigma_N} W_\sigma(X,Y) \quad \text{where} \quad W_\sigma(X,Y) = \sum_{i \in I} \|X_i - Y_{\sigma(i)}\|^2. \tag{2}$$

We note that the methods developed in this paper extend to arbitrary strictly convex distances such as $\|X_i - Y_{\sigma(i)}\|^p$ for $p > 1$. One can prove that W defines a metric on the set of discrete distributions $[X]$.

Linear program formulation. Computing this distance requires to compute the optimal assignment $i \mapsto \sigma^\star(i)$ that minimizes $W_\sigma(X,Y)$ in (2). It is possible to recast this problem as a linear programming one

$$W(X,Y)^2 = \min_{P \in \mathcal{P}_N} \sum_{i,j \in I^2} P_{i,j} \|X_i - Y_j\|^2 \tag{3}$$

where $\mathcal{P}_N$ is the set of bistochastic matrices, *i.e.* nonnegative matrices which rows and columns sum to 1. The problem (3) can be solved with standard linear programming algorithms and more dedicated methods in $O(N^{2.5}\log(N))$ operations (see [28]).

1D case. The case $d = 1$ has some special structure that allows for a much faster solution. Indeed, if one denotes by σ_X and σ_Y the permutations that order the points

$$\forall\, 0 \leqslant i < N-1, \quad X_{\sigma_X(i)} \leqslant X_{\sigma_X(i+1)} \quad \text{and} \quad Y_{\sigma_Y(i)} \leqslant Y_{\sigma_Y(i+1)} \tag{4}$$

the optimal permutation $\sigma^\star$ that minimizes (2) is

$$\sigma^\star = \sigma_Y \circ \sigma_X^{-1}, \tag{5}$$

so that point $X_{\sigma_X(i)}$ is assigned to the point $Y_{\sigma_Y(i)}$. The Wasserstein distance together with the optimal assignment can thus be computed in $O(N\log(N))$ operations using a fast sorting algorithm.

2.2 Sliced Wasserstein Distance

The computation of the Wasserstein distance W is however computationally too demanding for the application to image processing we have in mind, where N can be quite large. Moreover, W is too difficult to handle in problems requiring the optimization of point clouds with functional involving the Wasserstein distance.

For these reasons, we now consider an alternative metric between distributions denoted $\tilde{W}$, that is obtained by computing 1D Wasserstein distances of projected point clouds

$$\tilde{W}(X,Y)^2 = \int_{\theta\in\Omega} W(X_\theta, Y_\theta)^2 \mathrm{d}\theta \quad \text{where} \quad X_\theta = \{\langle X_i,\, \theta\rangle\}_{i\in I} \subset \mathbb{R} \tag{6}$$

where $\Omega = \left\{\theta \in \mathbb{R}^d \setminus \|\theta\| = 1\right\}$ is the unit sphere. We will refer from now to this metric as the *Sliced Wasserstein Distance*.

The sliced Wasserstein distance can be re-written using a series of 1D optimal assignments

$$\tilde{W}(X,Y)^2 = \int_{\theta\in\Omega} \min_{\sigma_\theta\in\Sigma_N} \sum_{i\in I} |\langle X_i - Y_{\sigma_\theta(i)},\, \theta\rangle|^2 \mathrm{d}\theta,$$

This alternative metric $\tilde{W}$ allows to use the special case of the 1D assignment, that can be solved in closed form easily using (5).

3 Barycenter in Wasserstein Space

3.1 Wasserstein Barycenter

Given a family $\{Y^j\}_{j\in J}$ of point clouds, we are interested in computing a weighted average point cloud $X^\star$, that is defined, by analogy to the Euclidean setting as the minimizer

$$\mathrm{Bar}(\rho_j, Y^j)_{j\in J} \in \operatorname*{argmin}_X E(X) = \sum_{j\in J} \rho_j W(X, Y^j)^2, \tag{7}$$

where $\rho_j \geqslant 0$, is a set of weights, that is constrained to satisfy $\sum_j \rho_j = 1$.

Except in the special case of normal distributions [29], there is no known closed form solution to the problem (7).

Independently to our work, Agueh and Carlier have performed a mathematical analysis of this problem [30]. They show the existence of a solution and a dual formulation for continuous distributions.

1D case. In the 1D case, the Wasserstein barycenter can be computed in $O(N\log(N))$ operations using the permutations σ_{Y^j} that order the sets of values $Y^j \subset \mathbb{R}$ as in (4). The barycenter then reads

$$\forall\, 0 \leqslant i < N, \quad \left(\mathrm{Bar}(\rho_j, Y^j)_{j\in J}\right)_i = \sum_{j\in J} \rho_j Y^j_{\sigma_{Y^j}(i)}.$$

This barycenter was used for texture mixing applications in [27]. This paper generalizes this approach to point clouds in arbitrary dimensions.

Sliced Wasserstein barycenter. Observe that in a way similar to (3), the optimization of (7) can be re-casted as a *linear program*, but this time involving a joint probability matrix of dimension $|J|$, see [31]. Solving such a problem requires to manipulate arrays of size $N^{|J|}$, and is therefore prohibitive even for small size problems.

We thus propose to replace the original Wasserstein metric by its sliced approximation $\tilde{W}$, and define a barycenter as a minimizer

$$\tilde{\mathrm{Bar}}(\rho_j, Y^j)_{j\in J} = \operatorname*{argmin}_X \tilde{E}(X) = \sum_{j\in J} \rho_j \tilde{W}(X, Y^j)^2 \tag{8}$$

3.2 Stochastic Gradient Descent Algorithm

Finding the barycenter by minimizing (8) corresponds to the minimization of a non-convex functional

$$\tilde{E}(X) = \int_{\theta\in\Omega} \tilde{E}_\theta(X)\mathrm{d}\theta \quad \text{where} \quad \tilde{E}_\theta(X) = \sum_{j\in J} W(X_\theta, Y^j_\theta). \tag{9}$$

It is possible to find a stationary point of this functional with a gradient descent algorithm.

Stochastic gradient descent. Since computing the descent direction with the whole set of directions $\theta \in \Omega$ is too expensive, we use a stochastic descent algorithm [32].

The algorithm starts from some initial point cloud $X^{(0)} \subset \mathbb{R}^d$, that can be for instance chosen to be any of the clouds Y^j, and sets $k = 0$. It picks at each iteration k a small set of random directions $\Omega_k \subset \Omega$. The size $|\Omega_k|$ of this set is fixed and finite, and is a parameter of the algorithm.

The descent update reads

$$X^{(k+1)} = X^{(k)} - \eta_k H_k^+ \sum_{\theta\in\Omega_k} \nabla \tilde{E}_\theta(X^{(k)})$$

where $\nabla \tilde{E}_\theta(X^{(k)})$ is the gradient of $\tilde{E}_\theta$ at point $X^{(k)}$ and $\eta_k > 0$ the step size. Here $H_k \in \mathbb{R}^{d\times d}$ is the Hessian matrix (or an approximation of it), and H_k^+ is the Moore-Penrose pseudo-inverse, which differs from the inverse if $|\Omega_k| < d$, since H_k is not invertible.

Gradient computation. For each $\theta \in \Omega_k$, computing the gradient $\nabla \tilde{E}_\theta(X^{(k)})$ requires, for each $j \in J$, to compute the optimal 1D assignment σ^j_θ that minimizes

$$\min_{\sigma^j_\theta \in \Sigma_N} \sum_{i\in I} |(X^{(k)}_\theta)_i - (Y^j_\theta)_{\sigma^j_\theta(i)}|^2.$$

This is computed as detailed in (5) by sorting the values $X^{(k)}_\theta$ and Y_θ, which requires $O(N\log(N))$ operations.

Each element of the gradient is expressed as

$$\nabla \tilde{E}_\theta(X^{(k)})_i = H_k X_i^{(k)} - \sum_{\theta\in\Omega_k, j\in J} \rho_j \langle Y^j_{\sigma^j_\theta(i)}, \theta\rangle \theta$$

where the Hessian matrix reads

$$H_k = \sum_{\theta\in\Omega_k} \theta\theta^{\mathrm{T}} = \left(\sum_{\theta\in\Omega_k} \theta_i\theta_j\right)_{0\leqslant i,j<d}.$$

Convergence of the algorithm. To ensure the convergence of the stochastic gradient descent to a local minimum $X^{(\infty)}$ of $\tilde{E}(X)$

$$X^{(k)} \overset{k\to+\infty}{\longrightarrow} \tilde{\mathrm{Bar}}(\rho_j, Y^j)_{j\in J}. \tag{10}$$

one should use a decaying gradient step size $\eta_k \sim 1/k^\alpha$ for $1/2 < \alpha \leqslant 1$, see [32]. Note that this barycenter actually depends on the initialization $X^{(0)}$ of the algorithm.

In practice, we do not find it necessary to use a decaying η_k. We always observe convergence to a local minimum with the fixed step size $\eta_k = 1$.

We note that in the special case of a single distribution $|J| = 1$, and when each Ω_k is a random orthogonal basis (so that $|\Omega_k| = d$), our algorithm is equivalent to the algorithm proposed in [3].

Numerical examples. Figures 1 and 2 show comparisons of Wasserstein barycenters and sliced approximations.

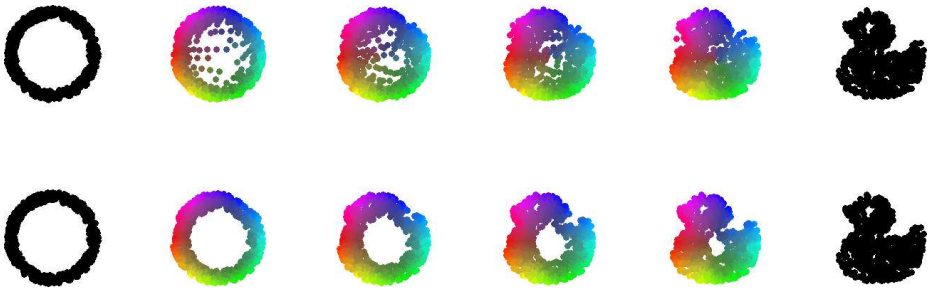


Fig. 1. Top row: sliced Wasserstein barycenters $\tilde{\mathrm{Bar}}(\rho, Y^1, 1-\rho, Y^2)$ for an increasing value of $\rho \in [0,1]$, for $N = 600$ points in $\mathbb{R}^2$. Bottom row: Wasserstein barycenters $\mathrm{Bar}(\rho, Y^1, 1-\rho, Y^2)$.

3.3 Computing the Projection on a Distribution

Projection on distribution constraints. In many applications, one is not only interested in computing the Wasserstein distance $W(X,Y)$, but also in the optimal permutation $\sigma^\star$ that minimizes $W_\sigma(X,Y)$ in (2). This optimal permutation

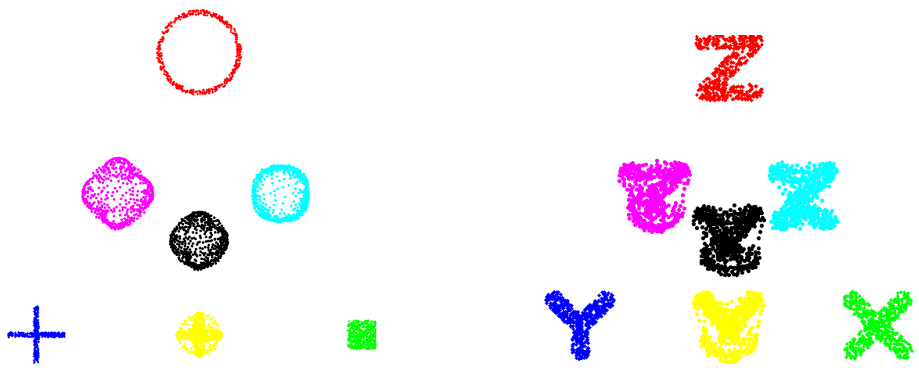


Fig. 2. Two examples (left and right panels) of barycenters $\tilde{\mathrm{Bar}}(\rho_j, Y^j)_{j=1,2,3}$. The top, bottom left, and bottom right corners of each triangle display respectively the distribution Y^1, Y^2, Y^3. The middle of each triangle edge display the barycenters for (ρ_1, ρ_2, ρ_3) equal to $(1/2, 1/2, 0)$, $(1/2, 0, 1/2)$ and $(0, 1/2, 1/2)$. The center of each triangle displays the barycenter for $(\rho_1, \rho_2, \rho_3) = (1/3, 1/3, 1/3)$.

allows one to compute the orthogonal projection

$$\mathrm{Proj}_{[Y]}(X)_i = X_{\sigma^\star(i)} \tag{11}$$

where the set $[Y]$ of all the point clouds that represent the same statistical distribution as Y is defined in (1).

This projector is at the heart of many statistical approaches for texture synthesis, that incorporate histogram constraints in there models, see for instance [18, 20]. These previous works consider only 1D histograms, so that the projection (11) is computed in $O(N \log(N))$ operations using (5). The remaining part of this section details how to perform approximate computations for higher dimensional distributions.

Sliced projection. Computing $\sigma^\star$ using linear programming (3) is computationally intractable in high dimensions. It is possible to compute an approximate projection by using the stochastic gradient descent algorithm described in section § 3.2 using a single distribution $\{Y^j\}_{j\in J} = \{Y\}$.

In this setting, the algorithm minimizes $\tilde{W}(X^{(k)}, Y)$, starting from $X^{(0)} = X$. A global minima $X^\star$ of this energy $\tilde{E}$ defined in (9) satisfies $\tilde{W}(X^\star, Y) = 0$ so that $X^\star \in [Y]$. The algorithm might however converge to a local minima $X^{(\infty)}$ with a non vanishing energy $\tilde{E}(X^{(\infty)}) > 0$, although this is rarely the case in practice.

We thus define the approximate projection of as the point clouds $X^{(\infty)}$ where $X^{(k)}$ is converging

$$\tilde{\mathrm{Proj}}_{[Y]}(X) = X^{(\infty)}. \tag{12}$$

Note that the mapping $\tilde{\mathrm{Proj}}_{[Y]}$ is not an orthogonal projection, although in practice it is closed to the orthogonal projection $\mathrm{Proj}_{[Y]}$.

Figure 3 shows on a 2D example that the sliced projection is in practice very close to the orthogonal one $\mathrm{Proj}_{[Y]}$.

Fig. 3. Left: initial distribution $X \subset \mathbb{R}^2$. Middle: sliced projection $\tilde{\mathrm{Proj}}_{[Y]}(X) = X^{(\infty)}$ defined in (12). Right: Wasserstein projection $\mathrm{Proj}_{[Y]}(X)$ defined in (11), computed by linear programming.

4 Texture Synthesis

This section applies the sliced Wasserstein barycenter (8)and the sliced Wasserstein projection (12)to perform texture mixing.

4.1 Multiscale Oriented Decompositions

We consider color textures exemplars $f^j \in \mathbb{R}^{P\times 3}$ of P pixels, where each pixel value is a 3D vector $f^j(x) \in \mathbb{R}^3$. A general framework for texture modeling makes use of the projection of the image on a set of atoms $\{\psi_{\ell,n}\}_{\ell\in L,n}$. All the atoms $\psi_{\ell,n} \in \mathbb{R}^P$ for a given $\ell \in L$ typically share a common scale and orientation, while n indexes a position.

Similarly to several previous works [18–20], we use a steerable wavelet tight frame [33]. In this case, the atoms $\psi_{\ell,n}$, where $\ell = (s, \theta)$, are parametrized by a dyadic scale 2^s (that indicates the size of the atoms), an orientation $\theta \in [0, \pi)$ and a position $2^s n \in [0, 1]^2$.

For the numerical experiments, we considered 4 dyadic scales and 4 orientations, together with a coarse scale frame (low pass residual) and a high frequency frame (details). The total number of frames in our experiments is thus $|L| = 4 \times 4 + 2 = 18$.

4.2 First Order Statistical Mixing

First order texture model. Following the work of Heeger and Bergen [18], the simplest texture model considers the first order statistics of the projection on the frames atoms. The model thus retains the distributions

$$\forall\, \ell \in L,\ \forall\, j \in J, \quad Y^{\ell,j} = \{\langle f^j,\, \psi_{\ell,n}\rangle\}_n.$$

All the models are computed in $O(|J|P\log(P))$ operations with the fast steerable pyramid transform [33]. Note that each coefficient $\langle f^j, \psi_{\ell,n}\rangle \in \mathbb{R}^3$ is a 3D vector obtained by projecting each channel of f^j onto the atom $\psi_{\ell,n}$.

First order texture mixing. Given a set $\{\rho_j\}_{j\in J}$ of weights, the first step of the mixing algorithm computes the barycentric model from the exemplar distributions, for both all scale and orientation $\ell \in L$ and for the pixel values

$$\forall\, \ell \in L, \quad Y^\ell = \tilde{\mathrm{Bar}}(\rho_j, Y^{\ell,j})_{j\in J} \subset \mathbb{R}^3 \quad \text{and} \quad Y = \tilde{\mathrm{Bar}}(\rho_j, f^j)_{j\in J} \subset \mathbb{R}^3$$

using the stochastic gradient descent algorithm detailed in Section 3.2 for $|J|$ distributions in $\mathbb{R}^3$.

Following [18], the synthesis of the mixed texture is then obtained by iteratively enforcing the statistical distribution using projections. The algorithm starts from a random white noise color image $f^{(0)} \in \mathbb{R}^{P\times 3}$. At iteration k, the algorithm computes the set of coefficients $\langle f^{(k)}, \psi_{\ell,n}\rangle$, and enforces the statistical constraints using the sliced projections

$$\forall\, \ell \in L, \quad \{c^{(k)}_{\ell,n}\}_n = \tilde{\mathrm{Proj}}_{[Y^\ell]}(\{\langle f^{(k)}, \psi_{\ell,n}\rangle\}_n) \tag{13}$$

using the stochastic gradient descent for only one distribution in $\mathbb{R}^3$. Since the steerable pyramid is a tight frame, an intermediate image is reconstructed in $O(P\log(P))$ operations as

$$\tilde{f}^{(k)} = \sum_{k\in K,n} c^{(k)}_{\ell,n}\psi_{\ell,n}. \tag{14}$$

The color pixel values distribution is then enforced as

$$\{f^{(k+1)}(x)\}_x = \tilde{\mathrm{Proj}}_{[Y]}(\{\tilde{f}^{(k)}(x)\}_x)$$

using once again the gradient descent method for a single distribution in $\mathbb{R}^3$.

Figures 4 and 5 shows the mixing of two and three textures using this first order model. Observe that this first order method generalizes the original method of Heeger and Bergen [18] by taking into account color distributions (whereas the original framework considers only 1D distributions obtained by a PCA change of color representation) and also by mixing several distributions.

4.3 Higher Order Statistical Mixing

Joint distribution model. The main limitation of the Heeger-Bergen method [18] and our extension is that it does not take into account the spatial correlation between wavelet coefficients. As a result, it yields poor synthesis results in the case of structured textures (see *e.g.* the brickwall texture in Figures 6(a) and 6(b)). Portilla and Simoncelli show in [20] that imposing self- and cross-correlation constraints between the scale and orientation of the steerable frames enables to

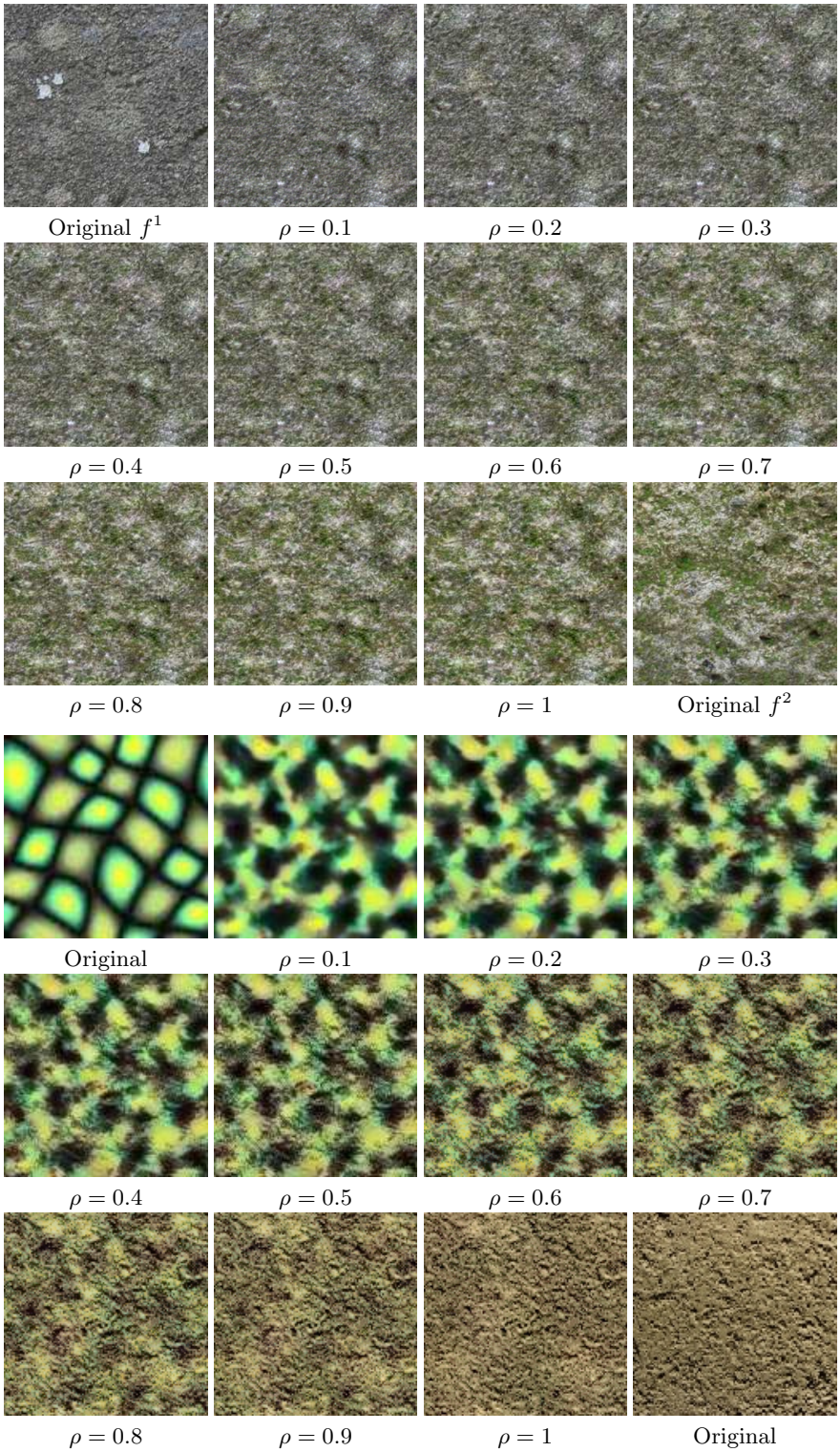


Fig. 4. Texture Interpolation : Mixing of two textures using the first order statistical model.

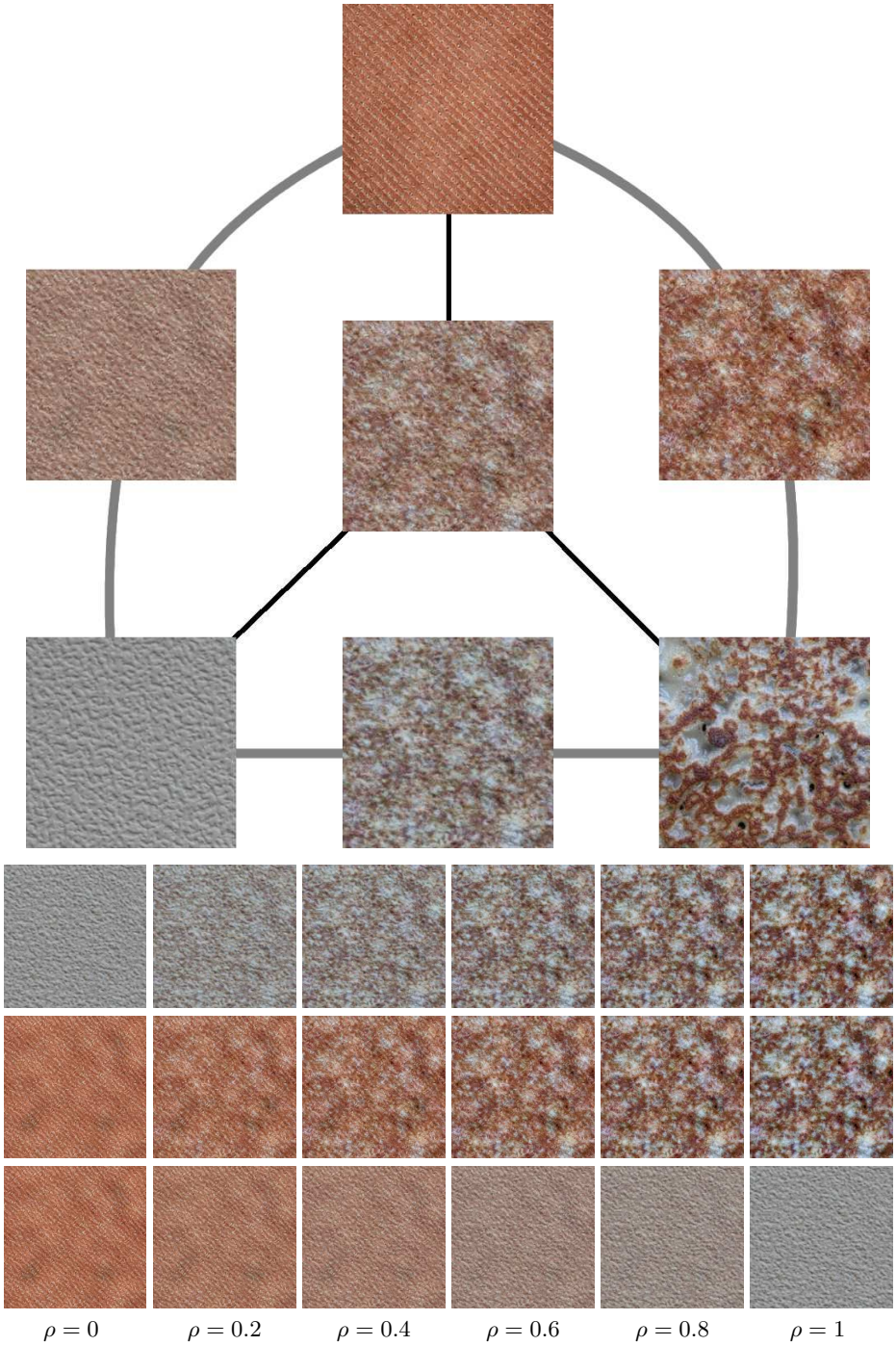


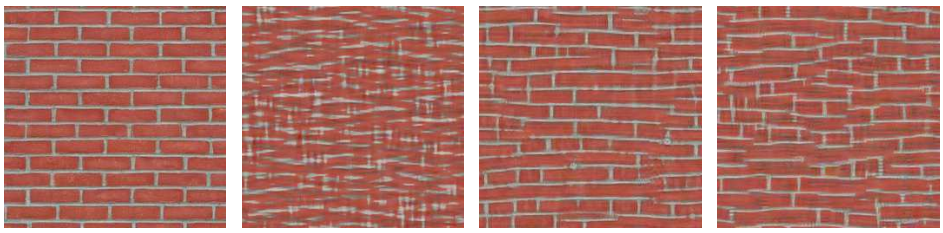
Fig. 5. Texture Interpolation of three textures using the first order statistical model. The three original textures are given at the vertices of the triangle (top Figure). Details of interpolations are given in the bottom.

preserve higher order statistical features in the texture synthesis process. An example is shown in Figure 6(c). The main limitations of this approach are firstly that the definition of the constraints are not fully generic, so that the projectors have to be designed by hand; secondly, the authors showed that it yields poor results for texture mixing.

This section extends the first order model described in Section 4.2 to handle higher order statistical features. Following the methodology introduced by Portilla and Simoncelli, we consider for each index ℓ – in addition to aforementioned the 3D-distributions $Y^{\ell,j}$ – the joint distributions of wavelet coefficients located in different spatial positions, *i.e.*

$$\forall\, \ell \in L,\ \forall\, j \in J, \quad C_{\mathcal{N}}^{\ell,j} = \{Y^{\ell,j}\{m\}, m \in \mathcal{N}(n)\}_n \subset \mathbb{R}^{3|\mathcal{N}|}.$$

where $\mathcal{N}(n) = n + \mathcal{N}$ is a given neighborhood pattern around each n. The joint distribution $C_{\mathcal{N}}^{\ell,j}$ thus has $3 \times |\mathcal{N}|$ dimensions. In the following experiments, $\mathcal{N}$ is defined as a square neighborhood of size 4×4, but more sophisticated pattern could be implemented. Figure 6(d) shows that the use of such joint-distributions allows to synthesize more complicated textures.



(a) Original (b) H-B. [18] (c) P-S. [20] (d) Our method

Fig. 6. (a) Original texture. (b) Heeger-Bergen texture synthesis (100 iterations). (c) Portilla-Simoncelli texture synthesis [20]with color constraint (100 iterations, using 5×5 pixel correlation constraints.). (d) Our method, with joint-distributions matching (100 iterations, using 4×4 blocks matching for each frame of the pyramid).

Higher order statistical texture mixing. The texture mixing framework introduced in section 4.2 is extended to take into account the joint distributions $\{C_{\mathcal{N}}^{\ell,j}\}_{j\in J}$. In a preliminary stage, the sliced Wasserstein barycenters $C_{\mathcal{N}}^{\ell} = \tilde{\mathrm{Bar}}(\rho_j, C_{\mathcal{N}}^{\ell,j})_{j\in J} \subset \mathbb{R}^{3|\mathcal{N}|}$ are computed using the stochastic gradient descent algorithm. Then, during the iterated synthesis algorithm, for each $\ell \in L$, the coefficients $\{c_{\ell,n}^{(k)}\}_n$ defined in (13) are clustered into blocks and projected into the statistical constraints

$$\forall\, \ell \in L, \quad \{\hat{c}_{\ell,n}^{(k)}\}_n = \tilde{\mathrm{Proj}}_{[C_{\mathcal{N}}^{\ell}]}(\{c_{\ell,m}^{(k)}, m \in \mathcal{N}(n)\}_n)$$

The images $f^{(k+1)}$ are then reconstructed from these coefficients $\{\hat{c}_{\ell,n}^{(k)}\}_n$ using (14).

Three different examples are proposed in Figure 7 to illustrate this high-order texture mixing framework. It should be noticed that the authors of [20] already introduced a color model, but that was not published. However, our method is the first extension of such joint probability constraints to texture mixing.

5 Conclusion

This paper has tackled the problem of defining average of histograms and distributions features. This shows that optimal distance metrics are useful beyond pairwise comparison of distributions.

The second point made by this paper is that the optimal transport in itself encompasses important information and enables to enforce complicated, high dimensional, statistical features. This paper presented an illustrative example in the field of texture synthesis.

Some extensions of this work are foreseen. To begin with, we do not have yet implemented cross block matching between different frames of the pyramid in our mixing algorithm as it is done in the original framework of Portilla and Simoncelli [20], this should improve the synthesis results visually. Moreover, more complex statistical information could be modeled using adaptive neighborhood $\mathcal{N}$, requiring a statistical learning step before the synthesis.

References

1. Villani, C.: Topics in Optimal Transportation. American Mathematical Society (2003)
2. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40** (2000) 99–121
3. Pitié, F., Kokaram, A., Dahyot, R.: Automated colour grading using colour distribution transfer. Computer Vision and Image Understanding (2007)
4. Dominitz, A., Tannenbaum, A.: Texture mapping via optimal mass transport. IEEE Transactions on Visualization and Computer Graphics **16** (2009) 419–433
5. Delon, J.: Movie and video scale-time equalization application to flicker reduction. IEEE Trans. Image Proc. **15** (2006) 241–248
6. Ambrosio, L., Caffarelli, L., Brenier, Y., Buttazzo, G., Villani, C.: Optimal Transportation and Applications. Mathematics and statistics edn. Volume 1813 of Lecture Notes in Mathematics. Springer (2003)
7. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. of ICCV '99. (1999) 1033
8. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proc. Siggraph '00. (2000) 479–488
9. Efros, A., Freeman, W.: Image quilting for texture synthesis and transfer. ACM Transactions on Graphics (2001) 341–346
10. Ashikhmin, M.: Synthesizing natural textures. In: SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics. (2001) 217–226
11. Lefebvre, S., Hoppe, H.: Parallel controllable texture synthesis. ACM Transactions on Graphics **24** (2005) 777–786

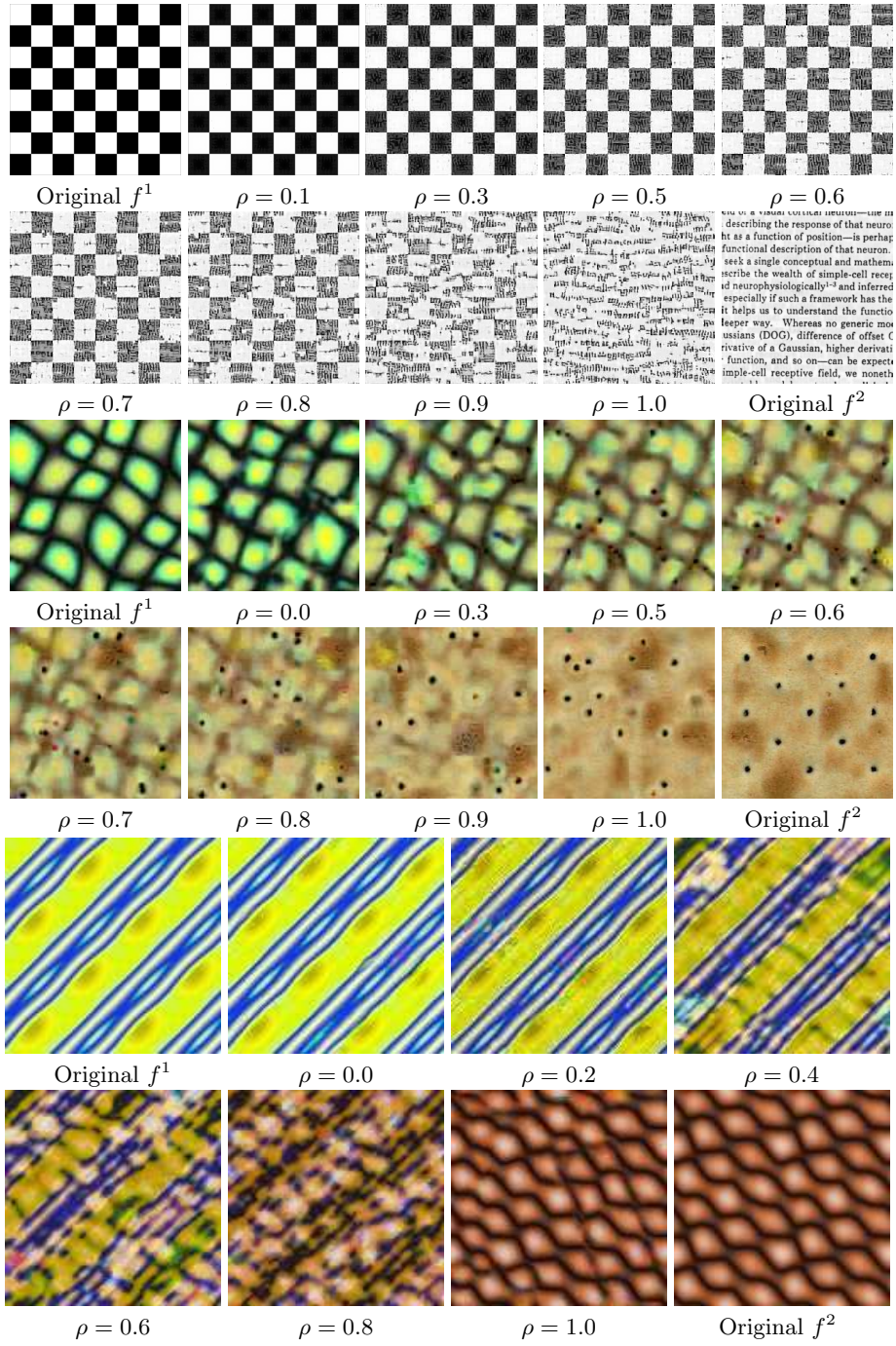


Fig. 7. High Order Texture Interpolation.

12. Kwatra, V., Essa, I., Bobick, A., Kwatra, N.: Texture optimization for example-based synthesis. ACM Transactions on Graphics **24** (2005) 795–802
13. Kwatra, V., Schdl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Transactions on Graphics **22** (2003) 277–286
14. Perlin, K.: An image synthesizer. In: Proc. Siggraph '85, New York, NY, USA, ACM Press (1985) 287–296
15. Bonet, J.S.D.: Multiresolution sampling procedure for analysis and synthesis of texture images. In: Proc. Siggraph '97, ACM Press/Addison-Wesley Publishing Co. (1997) 361–368
16. Paget, R., Longstaff, I.D.: Texture synthesis via a noncausal nonparametric multiscale markov random field. IEEE Trans. Image Proc. **7** (1998) 925–931
17. Mumford, D., Gidas, B.: Stochastic models for generic images. Q. Appl. Math. **LIV** (2001) 85–111
18. Heeger, D.J., Bergen, J.R.: Pyramid-Based texture analysis/synthesis. In: Proc. Siggraph '95. Annual Conference Series, ACM SIGGRAPH (1995) 229–238
19. Cook, R., DeRose, T.: Wavelet noise. ACM Transactions on Graphics **24** (2005) 803–811
20. Portilla, J., Simoncelli, E.P.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int. Journal of Computer Vision **40** (2000) 49–70
21. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. IEEE Transactions on Visualization and Computer Graphics **7** (2001) 120–135
22. Peyré, G.: Texture synthesis with grouplets. IEEE Trans. Patt. Anal. and Mach. Intell. **32** (2010) 733–746
23. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In ACM, ed.: Proc. Siggraph'01, ACM Press (2001) 327–340
24. Liu, Z., Liu, C., Shum, H.Y., Yu, Y.: Pattern-based texture metamorphosis. In: Proc. Pacific Graphics'02, IEEE Computer Society (2002) 184–193
25. Tonietto, L., Walter, M.: Texture metamorphosis driven by texton masks. Computers and Graphics **29** (2005) 697–703
26. Tal, A., Elber, G.: Image morphing with feature preserving texture. Comput. Graph. Forum **18** (1999) 339–348
27. Matusik, W., Zwicker, M., Durand, F.: Texture design using a simplicial complex of morphable textures. ACM Transactions on Graphics **24** (2005) 787–794
28. Burkard, R., Dell'Amico, M., Martello, S.: Assignment Problems. SIAM (2009)
29. Dowson, D.C., Landau, B.V.: The fréchet distance between multivariate normal distributions. J. Multivariate Anal. **3** (1982) 450?455
30. Agueh, M., Carlier, G.: Barycenters in the Wasserstein space. Unpublished manuscript (2010)
31. Gangbo, W., Swiechn, A.: Optimal maps for the multidimensional monge-kantorovich problem. Commun. on Pure and Appl. Math. **51** (1998) 23–45
32. Bottou, L.: Online algorithms and stochastic approximations. In Saad, D., ed.: Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK (1998)
33. Simoncelli, E.P., Freeman, W.T., Adelson, E.H., Heeger, D.J.: Shiftable multiscale transforms. IEEE Trans. Info. Theory **38** (1992) 587–607