



Optimal Patch Assignment for Statistically Constrained Texture Synthesis

Jorge Alberto Gutierrez Ortega, Julien Rabin, Bruno Galerne, Thomas Hurtut

► To cite this version:

Jorge Alberto Gutierrez Ortega, Julien Rabin, Bruno Galerne, Thomas Hurtut. Optimal Patch Assignment for Statistically Constrained Texture Synthesis. Scale Space and Variational Methods in Computer Vision. SSVM 2017, Jun 2017, Kolding, Denmark. 10.1007/978-3-319-58771-4_14 . hal-01510745

HAL Id: hal-01510745

<https://hal.science/hal-01510745>

Submitted on 19 Apr 2017

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.

Optimal Patch Assignment for Statistically Constrained Texture Synthesis

Jorge Gutierrez¹, Julien Rabin², Bruno Galerne³, and Thomas Hurtut¹

¹ Polytechnique Montréal, Montréal, Canada

² Normandie Univ., ENSICAEN, CNRS, GREYC, France

³ Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité
{jorge-alberto.gutierrez-ortega,thomas.hurtut}@polymtl.ca
julien.rabin@unicaen.fr bruno.galerne@parisdescartes.fr

Abstract. This article introduces a new model for patch-based texture synthesis that controls the distribution of patches in the synthesized texture. The proposed approach relies on an optimal assignment of patches over decimated pixel grids. This assignment problem formulates the synthesis as the minimization of a discrepancy measure between input’s and output’s patches through their optimal permutation. The resulting non-convex optimization problem is addressed with an iterative algorithm alternating between a patch assignment step and a patch aggregation step. We show that this model statistically constrains the output texture content, while inheriting the structure-preserving property of patch-based methods. We also propose a relaxed patch assignment extension that increases the robustness to non-stationnary textures.

Keywords: Example-based texture synthesis; patch matching; optimal assignment

1 Introduction

The goal of example-based texture synthesis is to generate a new texture image that reproduces the same visual characteristics as an input example without being an exact copy of it. One of the main issues is to simultaneously reproduce the global statistics of the example and the geometry of its local elements [16,15]. One can classify example-based texture synthesis methods into two categories, namely statistical matching methods and non-parametric patch-based methods.

Statistical matching methods estimate feature characteristics of the input texture to generate an output texture having the same characteristics. These features generally involve the input response to various filters whose distribution is summarized using either moments (mean value, correlation matrices, ...) or empirical distributions. In the literature, a broad range of such characteristics have been investigated such as color histograms, the Fourier modulus [10], steerable filter response histograms [14,6,22,26], and more recently occurrences in an adapted patch dictionary [27,28] or correlation matrices in convolutional neural

networks [11]. By essence, these methods are successful in preserving the characteristics of interest. They perform well on stochastic textures, yet they may fail to faithfully reproduce the local elements found in more structured textures.

Patch-based methods generate a new texture by sequentially copying pieces from the example [30]. First seminal works synthesize the output texture one pixel at a time [8,31] constraining the local coherence by choosing pixels with similar neighborhoods in the input. Subsequent contributions use patches instead of pixels as a unit of synthesis [21,7,18,20]. These methods usually better succeed at reproducing the input local structures. Their success is however not guaranteed: when the input image contains constant or blurry regions, these can be indeed enlarged during the synthesis, creating “garbage regions” [8,1,23]. A more principled approach consists in synthesizing the output texture through the minimization of a patch-based dissimilarity texture energy [19,13]. These methods are able to obtain high quality results for both stochastic and structured textures. Nevertheless, they still do not offer any guarantee of success, mostly because the energy does not convey any statistical matching. A first attempt to control the statistics of such patch-based synthesis is based on an empirical technique that compares the RGB histograms in order to penalize the overuse of a certain color in the output [17]. This histogram constraining technique helps to better employ the full color richness of the input. However, it requires a sequential pixel optimization, making it computationally expensive.

Over the last few years, numerical optimal transport has been shown to be a natural tool to solve efficiently optimization problems involving statistical constraints (e.g. [25]). Regarding texture synthesis, optimal transport has been used to solve the texture mixing problem using multi-histogram barycenters of steerable filter responses [26], and later for Gaussian textures [32].

The main purpose of this work is to show that numerical optimal transport is relevant for patch-based texture synthesis. Our insight for this claim is that the goal of patch-based texture synthesis algorithm is arguably to generate a new texture image whose patch distribution is the same as the input. Our main contribution is to use optimal assignment between the output and input patches. This results in a texture synthesis algorithm which enables both the reproduction of the local patterns of the input texture and the preservation of the global patch statistics (including color histogram). We also propose a relaxation of our model by allowing the assignment to vary from a permutation to an unconstrained nearest-neighbor matching.

The proposed model is first described in Section 2, and then the algorithm is detailed in Section 3 followed by an experimental validation and comparison with other methods in Section 4. We discuss a variant of the model in Section 5 in order to address some limitations.

2 Linear Patch Assignment Model

Notation Let $u_0 : \Omega_0 \rightarrow [0, 1]^3$ be the input example image and $u : \Omega \rightarrow [0, 1]^3$ be the output synthesized image, both using the normalized RGB color space.

To simplify the presentation, we first consider that the pixel grids $\Omega_0 \in \mathbb{Z}^2$ and Ω have the same size, i.e. $N = |\Omega| = |\Omega_0|$. We define $\mathcal{P}(x)$ as the set of indexes of pixels in the neighborhood of x which is a square of side w ($w = 8$ in our experiments)

$$\mathcal{P}(x) = \{x + t, t \in \{-\frac{w}{2}, \dots, \frac{w}{2} - 1\}^2\}$$

so that $p(x) = u \circ \mathcal{P}(x)$ is the patch of u in position x . The patches are extracted on sub-grids $\Omega_0^\downarrow$ and $\Omega^\downarrow$ of Ω_0 and Ω having a step size $\frac{w}{4}$ and size $N^\downarrow = |\Omega_0^\downarrow| = |\Omega^\downarrow|$. We use a symmetric boundary condition for patches of u_0 and periodic boundary condition for patches of u .

Proposed Model Our texture synthesis approach is driven by the following optimization problem which aims at minimizing the discrepancy between the patches from the synthesized texture and patches from the example image

$$\min_u \min_{\sigma \in \Sigma_{N^\downarrow}} \sum_{x \in \Omega^\downarrow} \|u \circ \mathcal{P}(x) - u_0 \circ \mathcal{P} \circ \sigma(x)\|_p^r \quad (1)$$

where $\sigma : \Omega^\downarrow \rightarrow \Omega_0^\downarrow$ is a mapping between the indexes of the patches. The most important aspect of this model is the definition of the set $\Sigma_{N^\downarrow}$ as the *set of permutations* of $N^\downarrow$ elements. The motivation is to synthesize a new texture that has the same distribution of patches as the example in order to preserve all its visual characteristics. We also consider $r = 1$ and the $\ell_{1,2}$ norm of patches, that is the sum of Euclidean norm of color coordinates

$$\|u \circ \mathcal{P}(x)\|_{1,2} = \sum_{y \in \mathcal{P}(x)} \|u(y)\|_2$$

These choices offer several advantages, resulting mainly in a separable convex optimization problem when minimizing with respect to u .

Connections with previous work Problem (1) is a generic framework for texture synthesis and it is closely related to several approaches from the literature such as [13, 17] inspired by the seminal work of [19]. For instance, Kwatra et al. [19] use the power $r = 0.8$ with an Euclidean norm weighted with a Gaussian falloff, and the assignment σ is not constrained to be a permutation, which results in a nearest neighbor matching. We claim that the absence of statistical constraint on the map σ is mainly responsible for the loss of features in the output textures, by discarding patches, and for the synthesis of “garbage” [8] and “blurry regions” [17], by locally repeating the same patches.

3 Algorithm

The motivation for Problem (1) is to find an image u which is pixel-wise different from the example image u_0 while having the same patch distribution. Yet Problem (1) has trivial solutions that are not relevant. For instance, using periodic boundary conditions, any circular shifting of the input image is a global

minimizer. Still, as usually done for variational texture synthesis [22, 19, 28], we minimize the non-convex functional of Problem (1) by alternate minimization with respect to u and σ separately starting from a random image. This relies on the assumption that local minimizers of the functional provide the desired result, which is verified in practice.

More precisely, Problem (1) is convex with respect to the image u for a given σ , *i.e.* when the assignment σ between patches has been fixed: this is the *patch aggregation* step that is addressed in § 3.1. In addition, the problem of finding an optimal assignment σ for a given synthesized image u is a solution of a relaxed convex problem: this is the *patch projection* step detailed in § 3.2. These two steps are sequentially used in a multi-scale scheme described in § 3.3.

3.1 Patch Aggregation

When the assignment σ between patches on the subgrids is fixed, optimizing the synthesized image u boils down to find

$$\operatorname{argmin}_u \sum_{x \in \Omega^\downarrow} \|u \circ \mathcal{P}(x) - u_0 \circ \mathcal{P} \circ \sigma(x)\|_{1,2} \quad (2)$$

This is a separable convex optimization problem: each color $u(x)$ is given by

$$\operatorname{argmin}_{u(x)} \sum_{y \in \mathcal{P}^\downarrow(x)} \|u(x) - u_0(\sigma(y) + x - y)\|_2 \quad (3)$$

For each pixel $x \in \Omega$, this simply corresponds to compute the color median of the 16 pixel values of patches overlapping x , located at

$$\mathcal{P}^\downarrow(x) = \{x + t, t \in \{-\frac{w}{2}, -\frac{w}{4}, 0, \frac{w}{4}\}^2\}$$

Our implementation solved the N problems (3) in parallel using a Douglas-Rachford splitting algorithm. Due to the lack of space, technical details are omitted and we refer to [5] for more information. We discuss in the experimental Section 4 the interest of this approach in comparison with other patch aggregation methods from the literature.

3.2 Optimal Patch Assignment

The output image u being fixed, an optimal assignment σ is solution of

$$\operatorname{argmin}_{\sigma \in \Sigma_{N^\downarrow}} \sum_{x \in \Omega^\downarrow} \|u \circ \mathcal{P}(x) - u_0 \circ \mathcal{P} \circ \sigma(x)\|_{1,2} \quad (4)$$

This problem can be recast as a *linear sum assignment problem* and solved in many different ways [4]. In practice, we use the Hungarian algorithm which is very fast for small assignment problems [3] (in our setting up to images with $N = 256^2$ pixels). For synthesizing larger images or to reduce the computation time, alternative methods should be considered such as parallel implementation of the auction algorithm [29] or the approximate assignment approach using the Sliced-Wasserstein distance [26].

Algorithm 1: Optimal patch assignment texture synthesis

Input: Example texture u_0
Parameters: Number of scales: $S = 3$, patch width: $w = 8$ px,
Iterations per scale: $\{I_s\}_{s=0}^{S-1} = \{10, 50, 50\}$
Initialization: $\{u_s\}_{s=0}^{S-1} \leftarrow$ Gaussian pyramid of u_0 (§3.3)
 $u \leftarrow$ Random image with same size as u_{S-1}
for $s = S - 1$ **to** 0 **do**
 for $i = 1$ **to** I_s **do**
 $\sigma \leftarrow$ Optimal assignment of patches of u to patches of u_s (§3.2)
 $u \leftarrow$ Patch aggregation using σ (§3.1)
 if $s \neq 0$ **then**
 $u \leftarrow$ Bilinear interpolation of u at scale $s - 1$ (§3.3)
Output: Synthesized image u

3.3 Multiscale Scheme and Initialization

A common way to capture large scale correlations between characteristics in the input is to use a coarse-to-fine synthesis [31, 18, 28]. We apply this strategy by computing the Gaussian pyramid $\{u_s\}_{s=0}^{S-1}$ of the input image, which is composed of S images u_s computed by filtering u_0 with a Gaussian kernel with standard deviation $0.8s$ and sub-sampled with a stepsize of 2^s .

The output coarser scale $s = S - 1$ can be initialized by any image. In practice, we use a random white noise image, as done in several variational texture synthesis previous works [14, 22, 27]. For subsequent scales, u is first upsampled by a factor 2 using bilinear interpolation. The resulting algorithm is described in Algorithm 1. In all experiments, the patch width is fixed to $w = 8$, the number of scales is $S = 3$, and the number of iterations I_s at scale s is $\{I_s\}_{s=0}^{S-1} = \{10, 50, 50\}$.

3.4 Discussion about Output Size

For simplicity, we have assumed up to this point that the output pixel grid Ω has the same size as the input grid Ω_0 , which results in defining σ as a permutation of the indexes of patches. However, in practice it is useful to synthesize output textures larger than the input. If $N^\downarrow = nN_0^\downarrow$ for some integer $n \geq 2$, a simple solution used in the experiments consists in duplicating n times the input patches. If $N^\downarrow$ is completely arbitrary, a usual solution is to simply crop a larger synthesized output. More relevant solutions might be obtained from bootstrapping, i.e. a random sampling of $N^\downarrow$ patches from the input distribution, or by considering multiple assignment. However, such a discussion is beyond the scope of this paper.

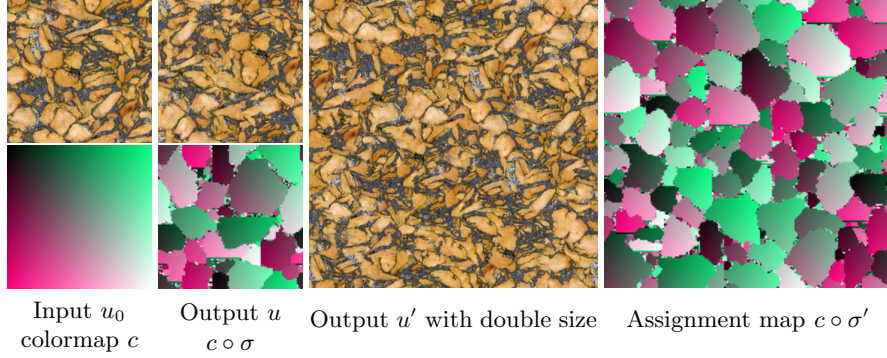


Fig. 1. Texture synthesis with single and double size outputs u and u' . A colormap c , shown bottom-left, is used to show the corresponding assignment mappings σ and σ' .

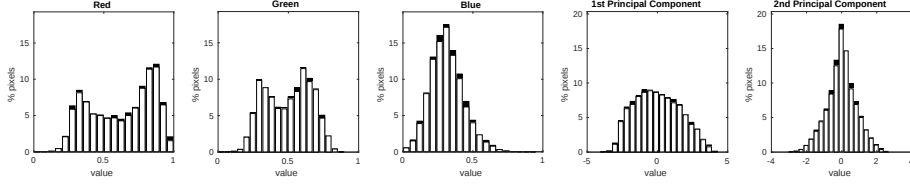


Fig. 2. RGB and patch histograms of the output u in Fig. 1. The absolute differences with the histograms from the input image u_0 are displayed in black.

4 Experimental Results and Comparison

Fig. 1 shows two synthesis results, the first having the same size as the input and the second having a double size, along with their respective assignment maps, that correspond to the last optimal assignment σ computed during the optimization process. One can observe with the double size output that the strict assignment forces to reproduce four times the structures and colors from the input. The assignment maps demonstrate that our approach, as all patch-based methods, tends to synthesize textures that are local verbatim copies of the example.

Statistics compliancy The patch assignment step generates a patch distribution that is identical to the one of the input. However the patch aggregation step produces new patches and therefore the input and output patch distributions are not strictly identical. Still, to assert experimentally that these two patch distributions are really close, we propose to compare the distributions of first-order statistics between the input and the output textures. In Fig. 2 are shown RGB per-channel color distributions and 1st and 2nd principal components distributions of patches using a PCA analysis on the input patch distribution. It illustrates, as one can observe by visual inspection of Fig. 1, that the features from the two images are mostly similar.

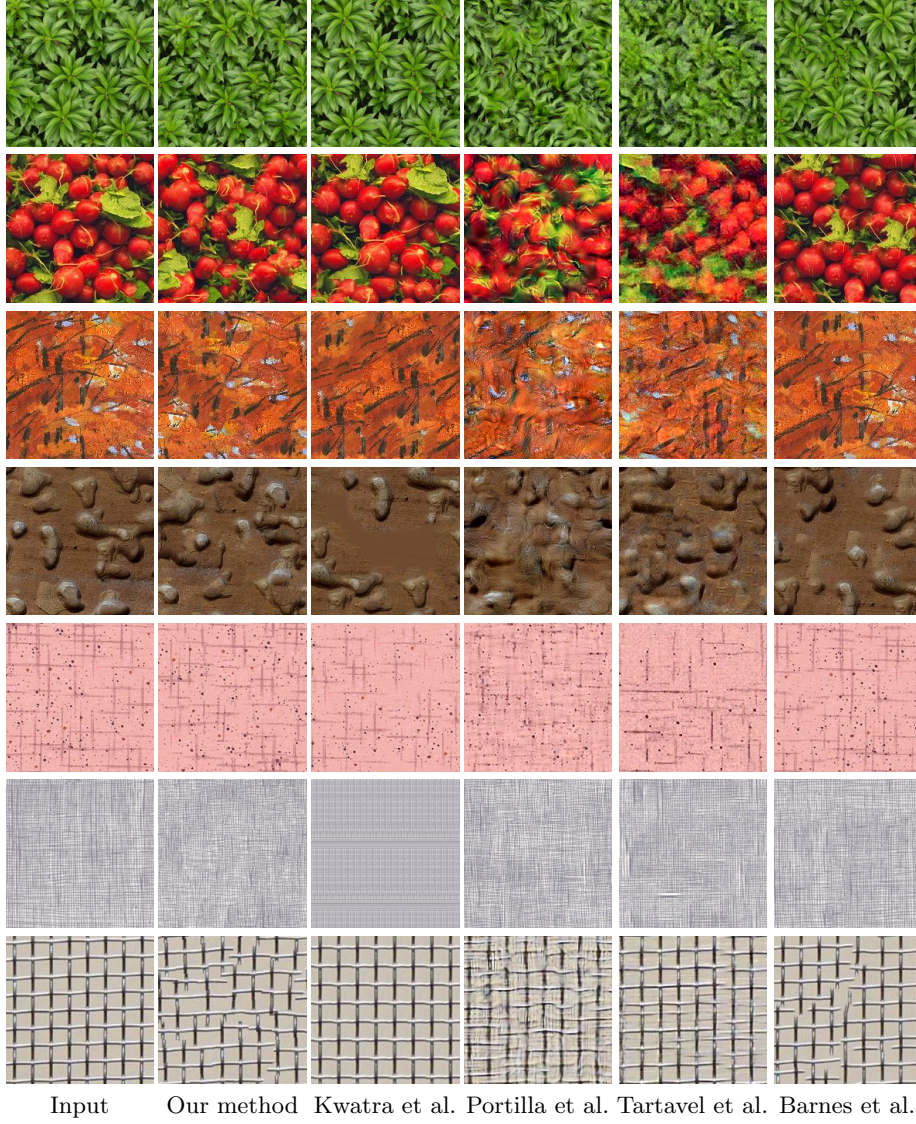


Fig. 3. Comparison of our approach with the patch-based method of Kwatra et al. [19], the two statistical matching methods of Portilla et al. [22] and Tartavel et al. [28], and the PatchMatch method of Barnes et al. [2] (with the implementation of David Tschumperlé for texture synthesis [12]). These approaches are used with the default parameters described in the respective papers.

Comparison with the state of the art Fig. 3 shows a comparison of the results obtained using our method with the patch-based approach of Kwatra et al. [19], and the statistical matching methods of Portilla et al. [22] and Tartavel et al. [28]. The two statistical matching methods successfully preserve the color distributions. However they often fail to faithfully reproduce local structured elements. Several issues can be noticed with Kwatra et al. results. Without any statistical constraints, this method sometimes tends to *simplify* the output content. First, copy of large areas from the input may be used (first and second rows). Second, low-cost patch combination tends to be favored: constant regions therefore tend to appear (fourth row), as well as simplified characteristics (loss of white strokes in the third example, and black spots in the fifth row). Last, when a small set of patches allows a periodic compositing, this set tends to be reused over and over (rows 6 and 7). The same issues are also raised with the non-variational state-of-the-art patch-based approach called PatchMatch [2] (see last column of Fig. 3). While our method is rather successful at both reproducing local elements and at preserving the global colors of the input, it has difficulty retaining long distance correlations on highly structured textures like in the last example of Fig. 3. The trade off between the number of scales used, the patches’ width, and the size of the input, limits the scale of the structures that can be captured. Thus, compared to using patches of multiple widths for each scale (as in Kwatra et al. [19]), our single-width approach loses long-distance correlations at a specific scale. However, this also means that smaller regions are copied from the input in comparison with [19], thereby avoiding repetitive artifacts (such as the shadows in the first example) and copy of large regions (second example).



Fig. 4. Comparison of results for different values of r from left to right: $r = 2$, $r = 0.8$, $r = 0.8$ with Gaussian falloff weighting and $r = 1$ using the same random initialization. Results are quite similar, except for $r = 2$ which is noticeably less sharp.

Discussion on patch aggregation As described in Section 3.1, once each patch of u is assigned to a patch u_0 , the image u is updated by minimizing the $\ell_{1,2}$ -norm, which results in computing the color median among the overlapping pixels. As a comparison, Kwatra et al. [19] use an ℓ_2 -norm (with Gaussian weights) to the power $r = 0.8$, and use an iteratively reweighted least squares scheme to minimize this functional (note that this kind of methods encounter numerical issues). Fig. 4 illustrates that our algorithm gives similar results when using this

variant, and that simply averaging by minimizing the squared ℓ_2 -norm leads naturally to blurry textures. We favored minimizing the $\ell_{1,2}$ -norm since it is a well-posed convex problem.

Limitations A limitation of our approach, due to the strict assignment constraint, is the guarantee to synthesize the same distribution of patches as in the input. Although this can be considered as an advantage in most situations, it may lead to undesired effects when inputs do not satisfy sufficiently the stationarity hypothesis.

5 Softening Statistical Constraints via Relaxed Patch Assignment

As demonstrated in the previous section, the synthesis of an image using the very same patch distribution as the one of the example image may lead to some limitations, typically when the stationarity hypothesis is false and the example contains some irrelevant features (e.g. due to illumination, scale change, or artifacts). We propose to address this problem by using soft statistical constraint in such a way that the input and output image may now have a close but different patch distribution.

5.1 Relaxed Assignment Model

We aim at defining a new synthesis model relying on a *relaxed* assignment of patches, that is, for which the one-to-one assignment constraint of Problem (1) is relaxed to enable multiple matching of some example patches. Such an idea was first proposed in [9] and refined by [24] to overcome the problem of color inconsistency in color transfer.

First, let us recall that an optimal assignment problem such as the one of Problem (4) can be recast as a Linear Sum Assignment Problem [4] of the form

$$\min_{\sigma \in \Sigma_{N^\downarrow}} \sum_{i \in \Omega^\downarrow} C_{i, \sigma(i)} = \min_{A \in \mathcal{A}} \sum_{(i,j) \in \Omega^\downarrow \times \Omega_0^\downarrow} A_{i,j} C_{i,j} \quad (5)$$

where C is a fixed cost matrix that corresponds to the distance between patches, that is, $C_{i,j} = \|u \circ \mathcal{P}(i) - u_0 \circ \mathcal{P}(j)\|_{1,2}$ and where

$$\mathcal{A} = \left\{ A \in [0, 1]^{N^\downarrow \times N^\downarrow}, \forall i \sum_j A_{i,j} = 1, \forall j \sum_i A_{i,j} = 1 \right\}$$

is the set of bistochastic matrices (which is the convex hull of the set of permutation matrices).

Now, following [24], we consider the relaxed assignment problem

$$\min_{P \in \mathcal{A}_k, k \geq 0} \sum_{i,j} P_{i,j} C_{i,j} + \rho \sum_j |k_j - q_j| \quad (6)$$

where the set of relaxed assignment matrix is defined as

$$\mathcal{A}_k = \{P \in [0, 1]^{N^\downarrow \times N^\downarrow}, \forall i \sum_j P_{i,j} = 1, \forall j \sum_i P_{i,j} = k_j\}. \quad (7)$$

In this model, q_j is the desired number of matches for the example patch indexed by j and k_j is the number of times this patch is actually matched. Experiments show that the solution of this problem is a relaxed assignment, and we leave the formal proof for future work. The relaxation parameter ρ controls the soft constraint on assignment: if $\rho = 0$, the problem boils down to a nearest-neighbor matching (as done in Kwatra et al. [19]), and for large enough values of ρ , the problem is the optimal assignment problem. To sum up, this model provides relaxed assignments that range from nearest neighbor matching to optimal assignment.

5.2 Results and Discussion

Fig. 5 illustrates the effect of relaxing the assignment at each iteration of Algorithm 1 using the above model. These experiments use the same random initialization and the vector q constant to 1. As expected, we observe that for large values of ρ the relaxed matching is an optimal assignment, while small values of ρ yields relaxed assignment so that some patches of the input example may not be used.

The main practical interest of this model is that it allows for discarding some patches that may represent a rare pattern in the texture (e.g. the red dots in the first example of Fig. 5 and the green leaf of the second example). Contrarily, some other features may now be replicated (the yellow dots in the first example, some radishes in the second). However, the parameter ρ does not offer an explicit control over the structures of the input image that may disappear. A solution to achieve this goal might be to let the user define q more precisely.

6 Conclusion

A new optimal assignment model has been proposed combining the advantages of statistically constrained methods and non-parametric patch-based texture synthesis algorithms. This model can be relaxed in order to be more controllable over the output distribution of patches. The visual quality of the results encourages future work regarding computation time acceleration as well as content-oriented control for the relaxation model.

References

1. C. Aguerrebere, Y. Gousseau, and G. Tartavel. Exemplar-based texture synthesis: the Efros-Leung algorithm. *IPOL*, 3:223–241, 2013.
2. C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAPH*, pages 24:1–24:11. ACM, 2009.

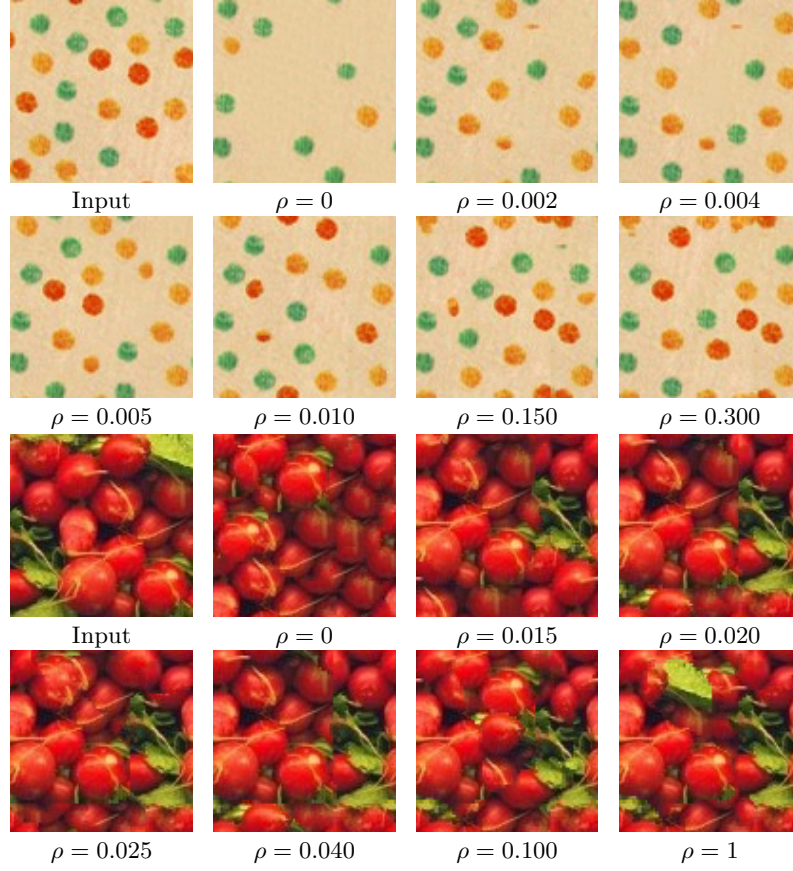


Fig. 5. Texture synthesis with relaxed assignments for different values of the relaxation parameter ρ . The value $\rho = 0$ corresponds to nearest-neighbor matching, which does not respect patch distribution. Increasing ρ gives relaxed assignment that allows for slight variation of the output patch distribution. For large values of ρ , the assignment is an optimal permutation resulting in the reproduction of characteristic of the input.

3. S. Bougleux and L. Brun. Linear sum assignment with edition. *preprint*, arXiv:1603.04380, 2016.
4. R. Burkard, M. Dell’Amico, and S. Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, 2012.
5. P. L. Combettes and J. C. Pesquet. A douglas rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, 2007.
6. J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *SIGGRAPH*, pages 361–368. ACM, 1997.
7. A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346. ACM, 2001.
8. A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, 1999.

9. S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol. Regularized discrete optimal transport. In *SSVM*, pages 428–439, 2013.
10. B. Galerne, Y. Gousseau, and J.-M. Morel. Random phase textures: Theory and synthesis. *IEEE Trans. Image Process.*, 20(1):257 – 267, 2011.
11. L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, pages 262 – 270. 2015.
12. G’MIC. Grey’s magic for image computing, 2016. <http://gmics.eu/>.
13. J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer*, 22(9):918–925, 2006.
14. D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238. ACM, 1995.
15. B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.
16. B. Julesz, E. N. Gilbert, and J. D. Victor. Visual discrimination of textures with identical third-order statistics. *Biological Cybernetics*, 31(3):137–140, 1978.
17. J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong. Solid texture synthesis from 2d exemplars. In *SIGGRAPH*. ACM, 2007.
18. V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *SIGGRAPH*, pages 277–286. ACM, 2003.
19. V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. In *SIGGRAPH*, pages 795–802. ACM, 2005.
20. S. Lefebvre and H. Hoppe. Parallel controllable texture synthesis. In *SIGGRAPH*, pages 777–786. ACM, 2005.
21. L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.*, 20(3):127–150, 2001.
22. J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.
23. L. Raad and B. Galerne. Efros and Freeman image quilting algorithm for texture synthesis. *IPOL*, 2016.
24. J. Rabin, S. Ferradans, and N. Papadakis. Adaptive color transfer with relaxed optimal transport. In *IEEE ICIP*, pages 4852–4856, 2014.
25. J. Rabin and G. Peyré. Wasserstein regularization of imaging problem. In *IEEE ICIP*, pages 1541–1544, 2011.
26. J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *SSVM*, pages 435–446, 2012.
27. G. Tartavel, Y. Gousseau, and G. Peyré. Constrained sparse texture synthesis. In *SSVM*, pages 186–197, 2013.
28. G. Tartavel, Y. Gousseau, and G. Peyré. Variational texture synthesis with sparsity and spectrum constraints. *JMIV*, 52(1):124–144, 2015.
29. C. N. Vasconcelos and B. Rosenhahn. Bipartite graph matching computation on gpu. In *EMMCVPR*, pages 42–55, 2009.
30. L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk. State of the art in example-based texture synthesis. In *Eurographics 2009 - State of the Art Reports*, 2009.
31. L. Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488. ACM, 2000.
32. G.-S. Xia, S. Ferradans, G. Peyré, and J.-F. Aujol. Synthesizing and mixing stationary Gaussian texture models. *SIAM J. on Imaging Science*, 8(1):476–508, 2014.