PACO: Signal Restoration via PAtch COnsensus

Ignacio Ramírez *Member, IEEE*

***Abstract*—Many signal processing algorithms operate by breaking the target signal into possibly overlapping segments (typically called windows or patches), processing them separately, and then stitching them back into place to produce a unified output. In most cases where pach overlapping occurs, the final value of those samples that are estimated by more than one patch is resolved by averaging those estimates; this includes many recent image processing algorithms. In other cases, typically frequency-based restoration methods, the average is implicitly weighted by some window function such as Hanning, Blackman, etc. which is applied prior to the Fourier/DCT transform in order to avoid Gibbs oscillations in the processed patches. Such averaging may incidentally help in covering up artifacts in the restoration process, but more often will simply degrade the overall result, posing an upper limit to the size of the patches that can be used. In order to avoid such drawbacks, we propose a new methodology where the different estimates of any given sample are forced to be identical. We show that, together, these consensus constraints constitute a non-empty convex feasible set, provide a general formulation of the resulting constrained optimization problem which can be applied to a wide variety of signal restoration tasks, and propose an efficient algorithm for finding the corresponding solutions. Finally, we describe in detail the application of the proposed methodology to three different signal processing problems, in some cases surpassing the state of the art by a significant margin.**

I. Introduction

Patch restoration refers to a family of methods where a signal, typically and image, is first broken down into smaller, possibly overlapping patches of some size and shape, some restoration method is applied to each patch separately, and finally the patches are stitched back together into the image to obtain a result. This is a common technique, with many examples in audio (e.g. [1], [2], [3], [4]) and image processing (see e.g., [5], [6], [7], [8], [9], [10], [11], [12], [13]). A recent review for the latter case can be found in [14]. The major drawback in most patch-based methods lies in their stitching phase, where the final value of a given sample is simply the average of all the recovered patches to which it belongs; we will refer to this stitching method as *Patch AVEraging* (PAVE) method hereafter. Applying PAVE generally results in a blurring effect. Furthermore, this blurring effect becomes larger with the size of the patches, thus limiting their maximum size in practice.

The patch blurring effect of PAVE is a well known problem to which many works have been devoted in recent years. Most of them [11], [12], [15], [16], [17] are based on patch *weighting* schemes, that is, they give more or less weight to each patch in the average. The work [18] also uses weighting, but does so at the single pixel level. On the other extreme, [19] uses a global formulation for optimizing the stitched image in terms of overall sparse representation.

Departamento de Procesamiento de Señales, Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, Uruguay.

A. Contributions

In contrast to the existing literature, this work does not propose any implicit or explicit weighting strategy. Instead, it enforces that the estimated patches *coincide exactly at their intersections*; this is generally known as a *consensus constraint*, hence we call our method *PAtch COnsensus* (PACO). Despite this constraint being conceptually very simple, we have not found any similar formulation in the literature, perhaps because it may seem overly strict at first look. As we will formally prove later, the consensus constraint is a non-empty linear feasible set whose dimension is equal to the length of the signal.

We then develop a general formulation for patch-based restoration problems under the PACO constraint which can be used applied to any pre-existing method that can be written as the minimization of a cost function of the estimated patches and/or signal. We propose a method for solving the aforementioned family of problems based on a splitting strategy and the standard ADMM [20] algorithm. This method has advantages of its own, such as allowing us to impose additional problem-specific constraints directly in signal space; in Section IV we present two algorithms that use this feature.

Although parallelization is not the main objective of this work, it turns out that the consensus strategy put forward here was developed, and is most commonly found in the literature about parallel and distributed computation (see [21] for a review.) It is thus a natural byproduct of our PACO framework to allow for parallel processing, making PACO an ideal framework for distributed processing of very large signals such as high resolution astronomical data, large 3D volumes, or audio, with the additional benefits that the PACO constraint brings in terms of restoration quality.

Finally, we show that PAVE can be interpreted as the first iterate of the PACO ADMM formulation, thus showing its sub-optimality with respect to the global optimum of the PACO optimization problem, which is guaranteed to be attained as long as the cost function and the constraint set are convex.

In summary, this paper makes the following contributions:

1) a formal mathematical framework for the patch stitching problem,
2) PACO, an optimal patch stitching strategy in the form of a family of optimization problems involving consensus constraints on the overlapping patches,
3) a general study of the feasibility and degrees of freedom of the PACO problem,
4) a general algorithm for efficiently solving instances of the aforementioned family of problems,

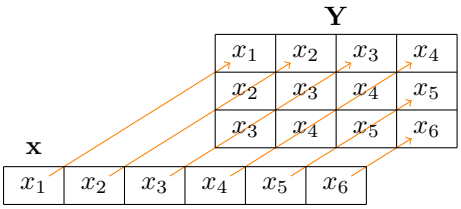

Fig. 1. Patch extraction for a one dimensional signal $\mathbf{x}$ of length $N = 6$ and patches of size $m = 3$; the patches are arranged as columns on an $m \times n$ matrix $\mathbf{Y}$ where the column $\mathbf{y}_k$ contains the patch starting at offset k in $\mathbf{x}$. Observe that $\mathbf{Y}$ is a Hankel matrix. This will be our running example.

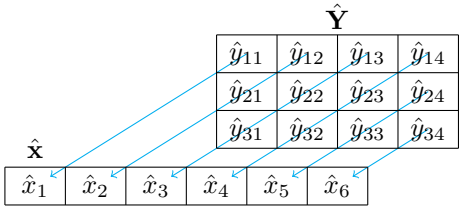

Fig. 2. Patch stitching operator. Each sample $\hat{x}_j$ is the average of all its estimates across the intervening patches. In terms of $\hat{\mathbf{Y}}$, this corresponds to the average of each anti-diagonal.

5) a formal proof showing that PAVE corresponds to the first step of our algorithm, thus proving its sub-optimality,
6) efficient implementations of the algorithm for missing data (inpainting) and/or denoising problems for both audio and 2D images. In the case of image inpainting, our results surpass the state of the art by a significant margin.

II. Background: Patch-based Signal Processing

A. Patch Extraction

Although the methods described hereafter are applicable to signals of any number of dimensions, the following discussion will be based on the one-dimensional case for simplicity. Such a signal is represented as $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N$; here $\mathbb{X} = \mathbb{R}^N$ is the space of all discrete finite signals of length N. Given $m \in \mathbb{N}$, $0 < m \leq N$, the maximum number of different contiguous patches of length m that we can extract from $\mathbf{x}$ is $n = N - m + 1$. This corresponds to the case where the starting index of any two contiguous patches in the signal differs by $s = 1$. This distance s is called the *stride* of the patch extraction process. Again, in order to keep the notation and the discussion simple, we will restrict ourselves to the case $s = 1$. However, the results developed hereafter are easily extended to strides larger than 1. In fact, in Section V we report on results obtained with our implementation for 2D images and values $s > 1$.

The patches are arranged as columns of a matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ so that the first column $\mathbf{y}_1$ corresponds to the patch $(x_1, x_2, \ldots, x_m)$, the second $\mathbf{y}_2 = (x_2, x_3, \ldots, x_{m+1})$, and so forth. The resulting *patches matrix* $\mathbf{Y}$ is shown for the case $N = 6$, $m = 3$ ($n = 4$) in Figure 1. As can be seen in Figure 1, with the proposed column ordering, the patch extraction procedure is a linear mapping from the signal space $\mathbb{X} = \mathbb{R}^N$ onto the space $\mathbb{H}$ of matrices whose anti-diagonals have a constant value (*Hankel* matrices) of size $m \times n$; we will denote this mapping as $\mathcal{E}$ so that $\mathbf{Y} = \mathcal{E}(\mathbf{x})$. Note that $\mathbb{H}$ is a linear subspace of $\mathbb{Y} = \mathbb{R}^{m \times n}$; we call $\mathbb{Y}$ the *patches space*.

The linear operator $\mathcal{E}(\mathbf{x})$ defines an isomorphism between the signal space $\mathbb{X}$ and $\mathbb{H}$. This implies in particular that $\dim(\mathbb{H}) = \dim(\mathbb{X}) = n$.

B. Patch-based Restoration

In a general signal restoration setting one does not observe the *true* or *clean* signal $\mathbf{x}$, but a distorted version $\tilde{\mathbf{x}}$ (which usually has the same size and dynamic range as $\mathbf{x}$), and the task is to infer $\mathbf{x}$ from $\tilde{\mathbf{x}}$. We denote the result of this inference as $\hat{\mathbf{x}}$, and call it the *restored* or *estimated* signal indistinctly. The matrix of patches extracted from $\tilde{\mathbf{x}}$ is correspondingly denoted by $\tilde{\mathbf{Y}}$. The idea is to estimate the signal $\mathbf{x}$ by recomposing it from an estimation of the clean patches, $\hat{\mathbf{Y}}$, which is a function of $\tilde{\mathbf{Y}}$. In general, however, $\hat{\mathbf{Y}} \notin \mathbb{H}$. A typical example where this occurs, and which we will deal with as a particular example later in Section IV-B, is the penalized regression problem. Here, each patch $\hat{\mathbf{y}}_j$ is inferred from $\tilde{\mathbf{y}}_j$ as follows

$$\hat{\mathbf{y}}_j = \mathbf{D}\hat{\mathbf{a}}_j\,,\ \hat{\mathbf{a}}_j = \arg\min_{\mathbf{a}} \frac{1}{2\tau}\|\mathbf{D}\mathbf{a} - \tilde{\mathbf{y}}_j\|_2^2 + \|\mathbf{a}\|_p^q,\ \forall\, j \quad (1)$$

where $\mathbf{D}$ is an $m \times p$ matrix, for example the matrix form of a linear operator such as the Discrete Fourier Transform (DCT). Each coefficients vector $\mathbf{a}_j \in \mathbb{R}^p$ defines the combination of columns of $\mathbf{D}$ that results in the estimated patch $\hat{y}_j$.

C. Patch Stitching

Once $\hat{\mathbf{Y}}$ has been computed, the final estimation $\hat{\mathbf{x}}$ must be recomposed from it; we call this procedure *patch stitching*. In our example, each column $\hat{\mathbf{y}}_j$ is an estimate of $(x_j, x_{j+1}, \ldots, x_{j+m-1})$. However, as patches overlap, each single signal sample (e.g., x_j) will be estimated many times, once for each patch whose mapping includes x_j. For example, if $m > 1$, we have that both $\hat{y}_{12}$ and $\hat{y}_{21}$ are estimates of x_2. In general, we have that x_j will be estimated once for each element in the j-th anti-diagonal of $\hat{\mathbf{Y}}$. The straightforward procedure in this case, followed by many successful restoration algorithms such as K-SVD [22], is to simply average all such estimations to produce the final result; this is what we defined as PAVE in Section I. Formally, the PAVE estimate at index j, $\hat{x}_j$ is the average of the values along the j-th anti-diagonal of $\hat{\mathbf{Y}}$; this is depicted in Figure 2. We note that this can be written as a non-invertible linear mapping $\mathcal{S} : \mathbb{Y} \to \mathbb{R}^n$, $\hat{\mathbf{x}} = \mathcal{S}(\hat{\mathbf{Y}})$.

D. Issues with PAVE stitching

In many patch-based restoration algorithms, the different patches are estimated independently of each other. In such cases, if m is too large, the averaging of many estimates can result in a blurred result, thus posing an upper limit on the practical size of m. This is evident in works such as [22], which are effective only for small patch sizes. The above problem can be alleviated by pre-multiplying each patch by a window (e.g., Gaussian, Blackman, Hamming, etc.) prior

to stitching. This is a "softer" way of stitching which also has theoretical advantages when combined with traditional frequency domain filtering, but it is not necessarily optimal in terms of the task at hand (e.g., in removing artifacts or blur).

The main contribution of this work arises from a simple premise: instead of estimating each patch independently, we force all patch estimations to coincide at the intersections, that is, we will seek solutions to our restoration problem that lie within the consensus set C defined by the problem. In this way, blurring is effectively eliminated regardless of the patch size. (Actually, there is no need for patch averaging, as all the estimates are identical at the optimum.) We will now describe our method in detail, its feasibility, and its exact resolution for a wide range of problems.

III. PACO: PATCH CONSENSUS

As mentioned in the previous section, our strategy is based on enforcing that all the estimated patches should coincide where they intersect. In the example we have been following so far, we require all feasible matrices $\hat{\mathbf{Y}}$ to be of the Hankel type. In general, we refer to the set of feasible matrices as the *patch consensus set* and denote it by C. An alternative representation of the set, again for one-dimensional signals, is given by,

$$C = \left\{ \begin{array}{l} \hat{y}_{j-i+1}[i] = \hat{y}_{j-i}[i+1], \\ \max\{1, j-n+m\} < i < \min\{j, m\} \\ j = 2, \ldots, n-1, \end{array} \right\}, \quad (2)$$

As discussed in II, the set C is a linear subset of $\mathbf{Y}$ of dimension equal to the dimension of the signal, n. It turns out, as we prove next, that projecting onto C can be performed very efficiently in terms of the stitching and extraction operations defined earlier.

Proposition 1 (Projection onto $\mathbb{H}$)**.** Let $\Pi_{\mathbb{H}}(\mathbf{A})$ be the projector operator from $\mathbb{Y}$ onto $\mathbb{H}$ in Frobenius norm,

$$\mathbf{A}^+ = \Pi_{\mathbb{H}}(\mathbf{A}) = \arg\min_{\mathbf{B}\in\mathbb{Y}} \|\mathbf{A}-\mathbf{B}\|_F^2.$$

Then $\mathbf{A}^+ = \Pi_{\mathbb{H}}(\mathbf{A}) = (\mathcal{E}\circ\mathcal{S})(\mathbf{A})$. (here $(f\circ g)(x) = f(g(x))$ denotes the composition of functions f and g.)

Proof. We can write $\|\mathbf{A}-\mathbf{B}\|_F^2 = \sum_{ij}(a_{ij}-b_{ij})^2$ and reorder this summation so that it is grouped along the anti-diagonals,

$$\|\mathbf{A}-\mathbf{B}\|_F^2 = \sum_{k=1}^{n} \sum_{l=\max\{1,k-n+m\}}^{\min\{k,m\}} \left[a_{l(k-l+1)} - b_{l(k-l+1)}\right]^2.$$

Since $\mathbf{A}^+ \in \mathbb{H}$ we have that $a^+_{l(k-l+1)} = h_k^+ \,\forall\, l$. As the problem is separable in the anti-diagonals, we can solve for each h_k independently,

$$h_k^+ = \arg\min_h \sum_{l=\max\{1,k-n+m\}}^{\min\{k,m\}} \left[a_{l(k-l+1)} - h\right]^2,$$

whose solution is the average of the k-th anti-diagonal.

Now, by the definition of the stitching operator $\mathcal{S}(\cdot)$, if $\mathbf{h} = \mathcal{S}(\mathbf{A})$ we have that $h_k = h_k^+$, $\forall k$. If we then apply the extraction operator $\mathcal{E}(\cdot)$ to $\mathbf{h}$ we get a matrix $\mathbf{A}^+ = \mathcal{E}(\mathbf{h})$ where each k-th anti-diagonal has a constant value $h_k^+ = h_k$. We have thus proven that $\mathbf{A}^+ = (\mathcal{E}\circ\mathcal{S})(\mathbf{A}) = \Pi_{\mathbb{H}}(\mathbf{A})$. □

The space of Hankel matrices $\mathbb{H}$ is a simple example of the consensus set C. It is easy to verify that the same result carries on to a more general case (n-dimensional signals, $s1$). In general, C is given by the span of the patch extraction operator co-domain, which we denote by $\mathrm{span}(\mathcal{E})$. Here we will only give a sketch the proof, which is a tedious extension of Proposition 1.

Proposition 2 (Projection onto C)**.** Consider a real-valued signal $\mathbf{x}$ defined over any discrete domain $\Gamma \subset \mathbb{Z}^M$,$M \geq 1$, and a pair of corresponding patch extraction operators $\mathcal{S}$ and $\mathcal{E}$, where by *corresponding* we mean that $(\mathcal{S}\circ\mathcal{E})(\mathbf{x}) = \mathbf{x}$. Consider the consensus set given by $C = \mathrm{span}(\mathcal{S})$. Then $\mathbf{A}^+ = \Pi_C = (\mathcal{E}\circ\mathcal{S})(\mathbf{A})$.

Proof. It is a well known result that projecting onto consensus sets is equivalent to replacing the discordant values by their average. Any extraction operator $\mathcal{E}$ defines a partition of the entries in the patches matrix $\mathbf{A}$ so that each group in the partition has a number of "copies" of some given sample of $\mathbf{x}$. Consider any patches matrix $\mathbf{A} \notin C$. The corresponding stitching operator $\mathcal{S}$ will, by definition, average the values of $\mathbf{A}$ in each group defined by $\mathcal{E}$ and place the result into the corresponding position in the signal. Then, the extraction operator $\mathcal{E}$ will extract that average and copy its value into all the elements belonging to the corresponding group of $\mathbf{A}$. Thus, the equality $\Pi_C = (\mathcal{E}\circ\mathcal{S})$ will always hold. □

A. Problem formulation

Let $c(\cdot)$ be the *convex indicator function* associated to the consensus subset C, $c : \mathbb{Y} \to \mathbb{R}\cup\{+\infty\}$,

$$c(\mathbf{Y}) = \left\{ \begin{array}{ll} 0 & , \quad \mathbf{Y} \in C \\ +\infty & , \quad \mathbf{Y} \notin C \end{array} \right. \quad (3)$$

In its most general form, the PACO restoration problem is given as follows,

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} f(\mathbf{Y}) + c(\mathbf{Y}) \text{ s.t. } \mathbf{Y} \in \Omega. \quad (4)$$

where the problem and algorithm dependent function $f(\mathbf{Y})$ measures the quality of $\mathbf{Y}$ as an estimation of the unobserved true patch matrix $\mathbf{Y}$, $c(\mathbf{Y})$ enforces the solution to lie within C, and Ω represents any additional constraint set imposed by the particular problem at hand.

As $c(\cdot)$ is a convex function, the problem (4) will also be convex if the function $f(\cdot)$ and the set Ω are convex. This is an enormous advantage compared to patch-based estimation problems which are non-convex and thus depend heavily on the initialization (e.g. [7]). In our case, the convexity in $f(\cdot)$ and Ω are sufficient guarantees to achieve global convergence. In Section V we will show how we can tackle some mainstream restoration problems using PACO, in some cases surpassing the state of the art.

B. Feasibility of PACO-based problems

A natural question about PACO is how much freedom is left to a restoration problem for obtaining a useful solution if the consensus constraint is imposed. The following proposition 3 provides an answer.

Proposition 3. Consider a signal of length n, decomposed in fully-overlapping patches of size m. Consider a matrix of estimated patches $\hat{\mathbf{Y}}$ of size $m\times(n-m+1)$. Assume w.l.o.g. that $2m \leq n$. The following properties hold:

i) The number of constraints imposed by the PACO consensus set is $(m-1)(n-m)$.
ii) For a linear problem where $\hat{\mathbf{Y}} = \mathbf{DA}$ and $\mathbf{D} \in \mathbb{R}^{m\times p}$ has full column rank, the number of degrees of freedom is $k = p(n-m+1) - (m-1)(n-m)$.
iii) In particular, if $p = m$ or the constraints are imposed directly on $\hat{\mathbf{Y}}$, the number of degrees of freedom is n, the length of the signal.

Proof. i) The first element x_1 is estimated by only one patch, so no restriction is imposed on $\hat{y}_{11}$. The second element is estimated by $\hat{y}_{12}$ and $\hat{y}_{21}$, which are constrained to be equal. This adds one linear constraint to the problem. The third element is estimated by three different values of $\hat{\mathbf{Y}}$ and so forth. Therefore, for $i = 0, \ldots, m-1$ we have $\sum_{i=0}^{m-1} i$ constraints. The exact same thing happens for the last m samples of the signal. The middle $n-2m$ samples involve $(m-1)$ constraints each. Adding these three constraint numbers in order we get:

$$\begin{aligned} k &= \textstyle\sum_{i=0}^{m-1} i + (n-2m)(m-1) + \sum_{i=0}^{m-1} i \\ &= 2\tfrac{(m-1)m}{2} + (n-2m)(m-1) \\ &= (m-1)(m+n-2m) = (m-1)(n-m). \end{aligned} \tag{5}$$

ii) For the linear model $\hat{\mathbf{Y}} = \mathbf{DA}$, the matrix $\mathbf{A}$ has $p\times(n-m+1)$ linear coefficients. From the previous item, the number of linear constraints is $k = (n-m)(m-1)$. Adding these two we get

$$p(n-m+1) - (n-m)(m-1).$$

iii) If $p = m$, the number of degrees of freedom will be

$$\begin{aligned} k &= (n-m+1)m - (n-m)(m-1) \\ &= (n-m)m + m - (n-m)m + (n-m) \\ &= m + n - m \\ &= n. \end{aligned}$$

The last case is trivially derived also from the fact that $\mathcal{E}$ defines an isomorphism between $\mathbb{X}$ and C, as discussed in Section II-B.

□

C. Numerical resolution

In the preceding subsection we defined the general form of a PACO-based restoration problem, showed it to be a convex problem with a linear non-empty convex consensus constraint (plus other possible constraints imposed by the specific task at hand). We now describe a simple and efficient method which can be applied to any such formulation. The method is based on the *proximal operator form* [21] of the popular *Alternating Directions Method of Multipliers* (ADMM) [20]. Let $f(\cdot)$ be any convex function. The proximal operator of $f(\cdot)$ with parameter λ is given by,

$$\operatorname{prox}_{\lambda f}(y) := \arg\min f(x) + \frac{1}{2\lambda}\|y-x\|^2 \tag{6}$$

The proximal operator has many interpretations. In particular, it can be seen as a generalization of the concept of gradient to non-differentiable functions. (See [21] and references therein). The ADMM method is an old method which is broadly applicable to a wide range of problems. Its proximal operator formulation simplifies the application of ADMM to non-differentiable functions. In particular, it is easy to check that the proximal operator of the convex indicator function $c(\cdot)$ of a set C is precisely the projection operator Π_C. This is particularly important in our case since, by means of Proposition 2, we can perform such projection efficiently when C is a patch consensus set.

What remains now is to reformulate (4) so that it can be solved using ADMM,

$$(\hat{\mathbf{Y}}, \hat{\mathbf{Z}}) = \arg\min f(\mathbf{Y}) + g(\mathbf{Z}) + \frac{1}{2\lambda}\|\mathbf{Y}-\mathbf{Z}\|_F^2 \text{ s.t. } \mathbf{Y} = \mathbf{Z}, \tag{7}$$

where $g(\mathbf{Z})$ is the indicator function of the set $C\cap\Omega$. Problem (7) is clearly equivalent to (4). The key difference is that (7) is separable in $\mathbf{Y}$ and $\mathbf{Z}$ and also strongly convex if $\lambda > 0$. The ADMM algorithm for (7) is given by,

$$\mathbf{Y}^{(t+1)} \leftarrow \operatorname{prox}_{\lambda f}\left(\mathbf{Z}^{(t)} - \mathbf{U}^{(t)}\right), \tag{8}$$

$$\mathbf{Z}^{(t+1)} \leftarrow \Pi_{C\cap\Omega}(\mathbf{Y}^{(t+1)} + \mathbf{U}^{(t)}), \tag{9}$$

$$\mathbf{U}^{(t+1)} \leftarrow \mathbf{U}^{(t)} + \mathbf{Y}^{(t+1)} - \mathbf{Z}^{(t+1)}. \tag{10}$$

Steps (8)–(10) are repeated until convergence is attained to within a specified tolerance. Step (10) is trivial and identical regardless of the function $f(\cdot)$. If $\Omega = \mathbb{Y}$, $g = c$ and Step (9) is given by,

$$\mathbf{Z}^{(t+1)} = (\mathcal{E}\circ\mathcal{S})\left(\mathbf{Y}^{(t+1)} + \mathbf{U}^{(t)}\right).$$

If $\Omega \subset \mathbb{Y}$ is convex, a general solution to step (9) can be obtained iteratively using Dykstra's Projection Algorithm [23]. In other cases, as we will see later, the solution can be found in closed form. Finally, Step (8) will depend on the form of $f(\cdot)$. In fact, any previously existing restoration method which can be formulated as the minimization of $f(\cdot)$ can be accommodated to the PACO framework by replacing this step with the corresponding solution. This in turn leads to the following crucial observation:

Proposition 4. Given a fitting function $f(\cdot)$ and a constraint set $\Omega \subseteq \mathbb{Y}$, the solution to (4) is optimal with respect to all patch-based methods that are defined in terms of f and Ω.

Proof. By its definition, the solution $\hat{\mathbf{Y}}$ to (4) minimizes f over $C\cap\Omega$. Let $\hat{\mathbf{W}}$ be a solution to $\arg\min f(\mathbf{W})$ s.t. $\mathbf{W} \in \Omega$. After stitching, the effective $\hat{\mathbf{W}}^+$ in use will be the projection of $\hat{\mathbf{W}}$ onto C. By the optimality of $\hat{\mathbf{Y}}$ we then have that $f(\hat{\mathbf{Y}}) \leq f(\hat{\mathbf{W}}^+)$. □

Aside from Proposition 4, note that, in general, there is no guarantee that $\hat{\mathbf{W}}^+$ is the projection of $\hat{\mathbf{W}}$ onto $C \cap \Omega$.

IV. Restoration using PACO

In this section we describe in detail two particular signal processing methods based on PACO.

A. PACO-DCT *inpainting*

The problem of *inpainting*, more generally known as *data completion*, is to estimate the unknown or missing samples of a signal $\mathbf{x}$ given that we do know its values values at a known subset of indexes $O \subset \{1, 2, \ldots, n\}$. In this case, the set of feasible solutions is specified in *signal* space, that is $\Omega \subset \mathbb{X}$, with $\Omega = \{\mathbf{z} : z_i = x_i, i \in O\}$. We seek for the estimate $\hat{\mathbf{X}} \in \Omega$ for which the DCT coefficients matrix of the corresponding patches $\hat{\mathbf{Y}}$, $\hat{\mathbf{Y}} = \mathbf{D}^\intercal \hat{\mathbf{Y}}$,[1] has minimum possible weighted ℓ_1 norm,

$$f(\mathbf{A}) = \sum_{ij} w_{ij} \|a_{ij}\|. \tag{11}$$

The corresponding PACO problem is given by,

$$\hat{\mathbf{A}} = \arg\min_{A} \sum_{ij} w_{ij} \|a_{ij}\| + g(\mathbf{DZ}) \quad \text{s.t.} \quad \mathbf{Z} = \mathbf{A}. \tag{12}$$

where $g(\mathbf{DZ})$ is the indicator function of $C \cap \Omega$. The idea of using the weighted ℓ_1 norm is to favor the most likely solution according to the commonly accepted fact that DCT coefficients follow a heavy tailed distribution, although with different scale parameter in each case (see [25] for an in-depth analysis). As the function $f(\mathbf{A})$ is separable in the elements of $\mathbf{A}$, so is its proximal operator, known as the *soft-thresholding* operator

$$\mathcal{T}_{\lambda_{ij}}(\cdot)(a_{ij}) = \min\{x + \lambda_{ij}, \max\{0, x - \lambda_{ij}\}\}$$

where $\lambda_{ij} = \lambda w_{ij}$.

An interesting *side effect* of PACO is that the projection onto the consensus constraint set C is a composition of two linear mappings, $\Pi_C(\mathbf{Y}) = (\mathcal{E} \circ \mathcal{S})(\mathbf{Y})$, the first going from patch space $\mathbb{Y}$ to signal space $\mathbb{X}$, and the second one going back from $\mathbb{X}$ to patch space $\mathbb{Y}$. Therefore, if the feasible subset Ω is a linear subspace of $\mathbb{X}$ as in this case, then the projection onto $C \cap \Omega$ can be efficiently obtained by applying the projection onto Ω "while in" $\mathbb{X}$, that is,

$$\Pi_{C \cap \Omega}(\hat{\mathbf{Y}}) = (\mathcal{E} \circ \Pi_\Omega \circ \mathcal{S})(\hat{\mathbf{Y}}). \tag{13}$$

The following pseudocode implements (13),

$$\begin{aligned} \mathbf{v} &\leftarrow \mathcal{S}(\mathbf{D}\mathbf{A}^{(t+1)} + \mathbf{U}^{(t)}) \\ v_i &\leftarrow x_i, \ \forall i \in O \quad (\text{this is } \Pi_\Omega) \\ \mathbf{Z}^{(t+1)} &\leftarrow \mathcal{E}(\mathbf{v}). \end{aligned}$$

As with (13), we can easily impose any number of additional constraints in signal space $\mathbb{X}$ as long as they correspond to convex subsets of that space. An examples of such constraint is the *clipping constraint*, which forces the estimated samples to lie within a valid range (e.g., 0–255 for grayscale images).

[1] For $\mathbf{D}$ we use the orthonormal variant of the DCT type II.

We note that, for a general matrix $\mathbf{D}$, the proximal operator of $g'(\mathbf{Z}) = g(\mathbf{DZ})$ may be hard to compute even if that of $g(\mathbf{Z})$ has a simple closed form. However, for unitary transforms such as the orthonormal DCT, we have that [21],

$$\text{prox}_{\lambda g'}(\mathbf{Z}) = \mathbf{D}^\intercal \text{prox}_{\lambda g}(\mathbf{DZ}). \tag{14}$$

We will describe how to tackle the non-orthonormal case later in Section IV-C.

The complete PACO-DCT inpainting algorithm consists of repeating the following steps until convergence:

$$\begin{aligned} a_{ij}^{(t+1)} &\leftarrow \mathcal{T}_{\lambda w_{ij}}(z_{ij}^{(t)} - u_{ij}^{(t)}), \ \forall i, j \\ \hat{\mathbf{Y}}^{(t+1)} &\leftarrow \mathbf{D}(\mathbf{A}^{(t+1)} + \mathbf{U}^{(t)}) \\ \hat{\mathbf{x}}^{(t+1)} &\leftarrow \mathcal{S}(\hat{\mathbf{Y}}^{(t+1)}) \\ \hat{x}_i^{(t+1)} &\leftarrow \hat{x}_i^{(t+1)}, \ \forall i \in \Omega \\ \mathbf{Z}^{(t+1)} &\leftarrow \mathbf{D}^\intercal \mathcal{E}(\hat{\mathbf{x}}^{(t+1)}) \\ \mathbf{U}^{(t+1)} &\leftarrow \mathbf{U}^{(t)} + \mathbf{A}^{(t+1)} - \mathbf{Z}^{(t+1)} \end{aligned} \tag{15}$$

Estimation of the weights: The success of (15) relies heavily on the weights $\mathbf{W} = \{w_{ij}\}$. Following the assumption that DCT coefficients follow a heavy-tailed distribution whose scale depends on the corresponding basis vector. One possibility is to have these coefficients pre-calculated. In our case, we run (15) a first time using $w_{ij} = 1$, and use the resulting matrix $\hat{\mathbf{Y}}$ to estimate $\mathbf{W}$ as follows,

$$w_{ij} = \frac{\epsilon}{\epsilon + (1/n)\sum_k A_{ik}}, i = 1, \ldots, n,$$

where ϵ is a parameter chosen by the user, which should be small compared to the average absolute value of the elements of $\mathbf{A}$. The preceding strategy can be repeated any number of times to further refine $\mathbf{W}$ and the final solution; we call this a *reweighting strategy*. In our experiments we do it at most once. Note that other alternatives exist, including assigning a different weight to each coefficient. This could yield better results, but would deserve a longer discussion which we cannot entertain here for lack of space (see [25]).

Choice of the penalty parameter λ: There is no general recipe for choosing the ADMM penalty parameter λ in (7) that works in all cases. However, by inspecting the terms that are at play (the cost function $f(\cdot)$ and the squared ℓ_2 term), we observe that the first scales as $n \times p \times \alpha^2$, where α is the dynamic range of the signal (typically $\alpha = 2^b - 1$ for b-bit digital samples.) In the weighted ℓ_1 case, $f(\cdot)$ scales as $n \times p \times \alpha$. Thus, we use instead

$$\lambda' = \frac{n \times p \times \alpha^2}{n \times p \times \alpha} = \lambda \alpha, \tag{16}$$

and adjust λ once for all problems in the same class.

B. Gaussian denoising using PACO

In this case we have $\tilde{\mathbf{x}} = \mathbf{x} + \eta$, were the elements of η are i.i.d. samples of a Gaussian distribution $\eta_i \sim \mathcal{N}(0, \sigma^2)$. We now show three alternative ways of using PACO for denoising under such hypothesis.

1) Projection onto ℓ_2 balls: A popular method for estimating the patches of $\hat{\mathbf{Y}}$ is given by,

$$\hat{\mathbf{y}}_j = \arg\min_{\zeta} f(\zeta) \;\text{ s.t. } \|\mathbf{D}\zeta - \tilde{\mathbf{y}}_j\|_2 \leq K\sigma. \tag{17}$$

Where K is chosen so that the ball $\mathcal{B}_{j,K\sigma} = \{\zeta : \|\mathbf{D}\zeta - \tilde{\mathbf{y}}_j\|_2 \leq K\sigma\}$ includes with high probability the unobserved clean sample $\mathbf{y}$. Under the i.i.d. Gaussian assumption, $\frac{1}{\sigma^2}\|\mathbf{y}_j - \tilde{\mathbf{y}}_j\|_2^2$ has a χ^2_m distribution. We then choose K so that

$$\begin{aligned} P\left(\|\mathbf{y}_j - \tilde{\mathbf{y}}_j\|_2 \leq K\sigma\right) &= P\left(\frac{1}{\sigma^2}\|\mathbf{y}_j - \tilde{\mathbf{y}}_j\|_2^2 \leq K^2\right) \\ &= F_{\chi^2_m}\left(K^2\right) = q \\ K &= \sqrt{F^{-1}_{\chi^2_m}(q)}. \end{aligned}$$

As each ball $\mathcal{B}_{j,K\sigma} \subset \mathbb{Y}$ is convex, so is the intersection $\mathcal{B}_{K\sigma} = \bigcap_j \mathcal{B}_{j,K\sigma}$. If we now define g to be the indicator function of $\mathcal{B}_{K\sigma} \cap C$ and use (11) as the cost function, we arrive at what we call the *patch ball denoising*. The drawback with this method is that the intersection between the constraint set and the patch ball needs to be computed iteratively using Dykstra's algorithm [23]. Nevertheless, in our experiments, convergence of the latter method is achieved in as little as two or three iterations.

An alternative approach similar to the one we used for inpainting is possible too: we can impose the estimated and observed *signals* to be close in ℓ_2 norm directly

$$\hat{\mathbf{Y}} = \arg\min_{\zeta} \sum_j \|\zeta_j\| \;\text{ s.t. } \|\mathcal{S}(\mathbf{D}\zeta) - \tilde{\mathbf{x}}\|_2 \leq K'\sigma. \tag{18}$$

Now the constraint set is a ball defined in signal space, $\mathcal{B}'_{K'\sigma} = \{\hat{\mathbf{x}} : \|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\| \leq K'\sigma\} \subset \mathbb{X}$; thus call this the *signal ball constraint*. Using g as the indicator function of $\mathcal{B}'_{K'\sigma} \cap C$ and using (11) as the cost function, we obtain what we call the *signal ball denoising* method. This method has at least two advantages. First, the projection is computed in closed form, as with the inpainting case by interleaving the projection onto $\mathcal{B}'_{K'\sigma}$ between $\mathcal{E}$ and $\mathcal{S}$. Also, for most signals $n \gg 100$ and $K' = \sqrt{F^{-1}_{\chi^2_n}(q)} \approx 1$ for a wide range of values of q, so that we can fix $K' = 1$. However, the method has one potential drawback: the constraint set might be too large or general in order to pinpoint good solutions. With this in mind, we define a third denoising variant which simply combines both constraint sets $\mathcal{B}_{K\sigma} \cap \mathcal{B}'_{K'\sigma}$, which leads to what we call the *double ball denoising* method and which gives the best results as reported in Section V. We will not write down the pseudocode in this case for lack of space; the interested reader can refer to the implementations provided along with this paper.

2) Penalized least squares: The function $f(\cdot)$ in this case is a weighted sum of (11) and a fitting term which measures how close the estimated patches $\hat{\mathbf{Y}}$ are to the observed patches $\tilde{\mathbf{Y}}$ in Frobenius norm,

$$f(\hat{A}) = \frac{1}{2\tau}\|\mathbf{DA} - \tilde{\mathbf{Y}}\|_F^2 + \sum_{ij} w_{ij}\|a_{ij}\|, \tag{19}$$

where τ is a scalar parameter. A typical choice is $\tau = \sigma^2$. By adding the PACO indicator function $c(\mathbf{DA})$, we obtain the PACO problem for this case,

$$\begin{aligned} \hat{\mathbf{A}} &= \arg\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{DA} - \tilde{\mathbf{Y}}\|_F^2 + \sum_{ij} w_{ij}\|a_{ij}\| + c(\mathbf{DZ}) \\ \text{s.t.} &\quad \mathbf{Z} = \mathbf{A}. \end{aligned} \tag{20}$$

Note that (20) admits the alternative splitting

$$\begin{aligned} \hat{\mathbf{A}} &= \arg\min_{\mathbf{A}} \sum_{ij} w_{ij}\|a_{ij}\| + c(\mathbf{DZ}) + \frac{1}{2}\|\mathbf{Z} - \tilde{\mathbf{Y}}\|_F^2 \\ \text{s.t.} &\quad \mathbf{Z} = \mathbf{A}. \end{aligned} \tag{21}$$

Which splitting is more practical will depend on the case. For unitary $\mathbf{D}$, we know from (22) that the proximal operator of $c'(\mathbf{Z}) = c(\mathbf{DZ})$ is straightforward to compute. It turns out that the same happens with $f(\cdot)$ in this case. For this we apply the definition of the proximal operator of f with parameter λ,

$$\begin{aligned} \text{prox}_{\lambda f_d}(\mathbf{B}) = &\quad \arg\min_{\mathbf{A}} \Big\{ \textstyle\sum_{ij} w_{ij}|a_{ij}| + \\ &\quad \tfrac{1}{2\tau}\|\mathbf{DA} - \tilde{\mathbf{Y}}\|_F^2 + \tfrac{1}{2\lambda}\|\mathbf{A} - \mathbf{B}\|_F^2 \Big\} \\ = &\arg\min_{\mathbf{A}} \Big\{ \textstyle\sum_{ij} w_{ij}|a_{ij}| + \\ &\tfrac{1}{2\tau}\|\mathbf{A} - \mathbf{D}^\intercal\tilde{\mathbf{Y}}\|_F^2 + \tfrac{1}{2\lambda}\|\mathbf{A} - \mathbf{B}\|_F^2 \Big\}. \end{aligned} \tag{22}$$

So, again, the problem is separable in the elements of $\mathbf{A}$. The proximal operator in this case is given by (see Appendix A for its derivation),

$$a_{ij} = \mathcal{T}_{\theta_{ij}}\left(\frac{\lambda}{\lambda+\tau}\tilde{v}_{ij} + \frac{\tau}{\lambda+\tau}b_{ij}\right),$$

where $\tilde{v}_{ij}$ is the (i,j)-th element of the matrix $\tilde{\mathbf{V}} = \mathbf{D}^\intercal\tilde{\mathbf{Y}}$, that is, the DCT transform of the observed patches, and the threshold parameter is given by

$$\theta_{ij} = \frac{\lambda\tau w_{ij}}{\lambda+\tau}.$$

Note that $\tilde{\mathbf{V}}$ needs to be computed only once. In our case, the auxiliary matrix $\mathbf{B}$ at iteration t is given by $\mathbf{B}^{(t)} = \mathbf{Z}^{(t)} - \mathbf{U}^{(t)}$ (this matrix needs not be explicitly computed or stored; we include it as an intermediate step for clarity). The steps of the ADMM algorithm for PACO-DCT denoising are as follows:

$$\begin{aligned} \mathbf{B}^{(t)} &\leftarrow \mathbf{Z}^{(t)} - \mathbf{U}^{(t)} \\ a_{ij}^{(t+1)} &\leftarrow \mathcal{T}_{\theta_{ij}}\left(\frac{\lambda}{\lambda+\tau}\tilde{v}_{ij} + \frac{\tau}{\lambda+\tau}b_{ij}^{(t)}\right),\ \forall\, i,j \\ \hat{\mathbf{Y}}^{(t+1)} &\leftarrow \mathbf{D}(\,\mathbf{A}^{(t+1)} + \mathbf{U}^{(t)}\,) \\ \mathbf{Z}^{(t+1)} &\leftarrow \mathbf{D}^\intercal\mathcal{E}[\,\mathcal{S}(\hat{\mathbf{Y}}^{(t)})\,] \\ \mathbf{U}^{(t+1)} &\leftarrow \mathbf{U}^{(t)} + \mathbf{A}^{(t+1)} - \mathbf{Z}^{(t+1)}. \end{aligned}$$

A typical initialization would be $\mathbf{U}^{(0)} = 0$, $\mathbf{Z}^{(0)}$ some simple approximation to the input signal (for example, filling in the missing samples by the average of the known samples), and $\mathbf{A}^{(0)} = 0$. As long as the algorithm is convex, the final result will not depend on the initialization, although the speed of convergence can vary significantly depending on this.

C. PACO *restoration for non-orthonormal linear operators*

When $\mathbf{D}$ is non-orthonormal, a variant of ADMM known as "Linearized ADMM" (LADMM) or "inexact Uzawa's Method" [24] is more adequate for solving (4). Essentially, this method constructs the augmented Lagrangian for the constraint $\mathbf{Z} = \mathbf{DA}$ and then solves a linear approximation of it around $\mathbf{Z}$ in each iteration. Global convergence is guaranteed as long as its parameter μ obeys $0 < \mu \leq \lambda/\|\mathbf{D}\|_2^2$. The general method is given by,

$$\mathbf{A}^{(t+1)} \leftarrow \mathrm{prox}_{\mu f}\left(\mathbf{A}^{(t)} - (\mu/\lambda)\mathbf{D}^\intercal(\mathbf{DA}^{(t)} - \mathbf{Z}^{(t)} + \mathbf{U}^{(t)})\right)$$
$$\mathbf{Z}^{(t+1)} \leftarrow \mathrm{prox}_{\lambda c}\left(\mathbf{DA}^{(t+1)} + \mathbf{U}^{(t)}\right)$$
$$\mathbf{U}^{(t+1)} \leftarrow \mathbf{U}^{(t)} + \mathbf{DA}^{(t+1)} - \mathbf{Z}^{(t+1)}.$$

The corresponding LADMM PACO algorithm is given by

$$\mathbf{A}^{(t+1)} \leftarrow \mathrm{prox}_{\lambda f}\left(\mathbf{A}^{(t)} - (\mu/\lambda)\mathbf{D}^\intercal(\hat{\mathbf{Y}}^{(t)} - \mathbf{Z}^{(t)} + \mathbf{U}^{(t)})\right) \quad (23)$$
$$\hat{\mathbf{Y}}^{(t+1)} \leftarrow \mathbf{DA}^{(t+1)} + \mathbf{U}^{(t)} \quad (24)$$
$$\mathbf{Z}^{(t+1)} \leftarrow \mathcal{E}[\,\mathcal{S}(\hat{\mathbf{Y}})\,] \quad (25)$$
$$\mathbf{U}^{(t+1)} \leftarrow \mathbf{U}^{(t)} + \hat{\mathbf{Y}}^{(t+1)} - \mathbf{Z}^{(t+1)}. \quad (26)$$

The LADMM method is usually slower than ADMM. Besides, in the DCT case, we can compute $\mathbf{Dx}$ and $\mathbf{D}^\intercal\mathbf{x}$ in $O(m \log_2 m)$ operations, whereas for general $\mathbf{D}$ the number of operations is $O(mp) \approx O(m^2)$, which also slows down each iteration. Nevertheless, this allows for a much wider choice of $\mathbf{D}$, including wavelets and adaptive dictionaries, which can result in significantly better results in many applications.

V. Experimental results and discussion

Here we present results of two types. The first set of results study the rate of convergence of the PACO algorithm to the global solution with respect to the ADMM optimization parameters. The second set of results demonstrate the effectiveness of PACO on three settings: image inpainting, audio inpainting, and image denoising.[2]

We use two quality metrics to measure the performance of our algorithms. One is the root mean squared error, $\mathrm{RMSE} = n^{-1/2}\|\mathbf{x} - \hat{\mathbf{x}}\|_2$.[3] The other is the *Structural Similarity Index* (SSIM) [26], a well established metric designed to reflect the visual quality of the results; this is important in modern signal restoration, as measures such as RMSE or PSNR alone can be misleading in terms of the visual quality of the results. The SSIM ranges between 0 and 1; larger values indicate better quality.

A. Convergence analysis

The convergence results shown in Figure 4 are for the image inpainting problem on a small artificial test image shown in Figure 3; the image is shown at the bottom right. Note that the best result in Figure 3 is significantly better than the optimum obtained in our convergence tests, which correspond to the first run of the of the algorithm, with equal weight on all DCT coefficients. This leads to a significant (although visually unnoticeable) bias which impacts the RMSE but not the perceptual SSIM quality metric.

Figure 4 shows how the ADMM algorithm evolves for different penalty parameters λ as defined in (16). As expected, the algorithm converges in all cases, showing the typical ADMM behavior depending on the value of λ: for too large λ the main and split iterates tend to oscillate, which can

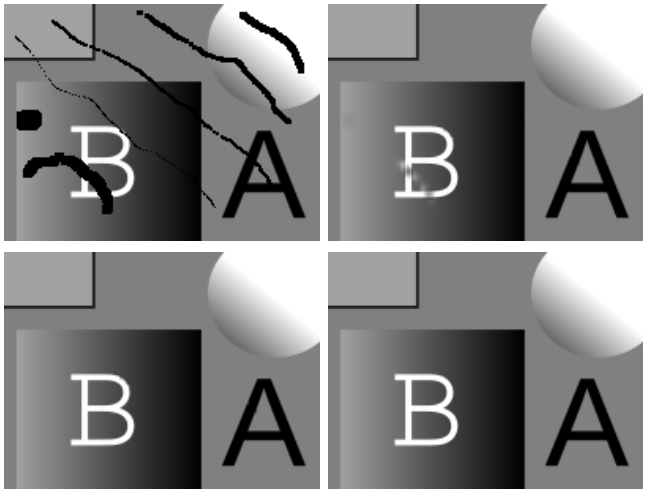

Fig. 3. Inpainting result on artificial test image. Top to bottom, left to right: input image; result after 10th iteration; final output RMSE = 0.83, SSIM = 0.99997. This was obtained using 10×10 patches, $\lambda = 0.3$ and one reweighting iteration.

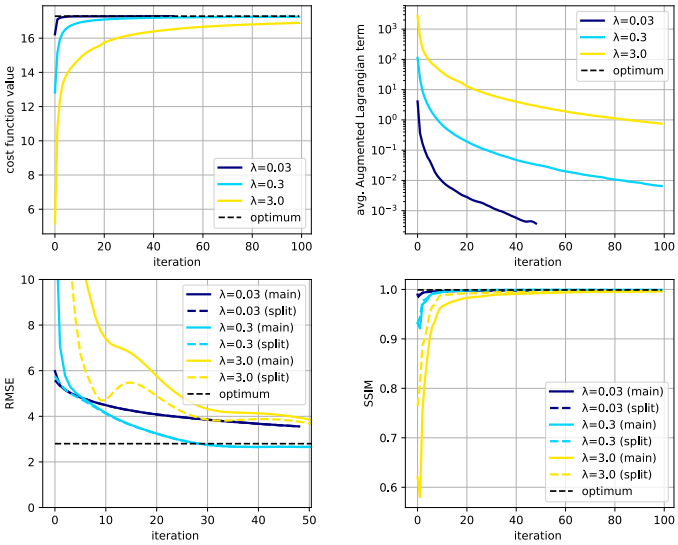

Fig. 4. Convergence analysis of the inpainting test problem shown for different fixed values of the parameter λ. We show, from top to bottom and left to right, the evolution of the values of: the (restoration) cost function $f(\cdot)$, the quadratic penalty term, the RMSE of the recovered images (solid for that derived from the main variable and dashed for the one derived from the split variable), and the corresponding SSIM index. The optimum value in each case (which was estimated by setting a much smaller tolerance on the stopping condition, after many more iterations) is shown as a dashed black line; note that, by definition, this value is 0 for the Lagrangian term.

[2]We provide additional results, as well as C++ and Python implementations of PACO in the supplementary material and on the project web page `http://iie.fing.edu.uy/~nacho/paco/`.

[3]The classic PSNR metric is not included as it is redundant with the RMSE; the interested reader can derive it as $\mathrm{PSNR} = 20\log_{10}(\alpha/\mathrm{RMSE})$ where $\alpha = 2^b - 1$; in our case $b = 8$ for images and b=16 for audio.

TABLE I
SUMMARY OF INPAINTING RESULTS ON THE KODAK DATASET. WE COMPARE OUR RESULTS WITH THOSE OF [27] AND [28] IN TERMS OF THE THE 25, 50 (MEDIAN) AND 75 PERCENTILES OF THE RMSE AND SSIM SCORES ON ALL 24 IMAGES. AS CAN BE SEEN, WE OBTAIN SIGNIFICANTLY LOWER RMSES AND HIGHER SSIM IN ALL CASES. THE CLOSEST RESULT IS THE MEDIAN RMSE OBTAINED WITH [27] ON MASK #2, WHICH COINCIDES WITH OURS. NOTE HOWEVER THAT THE MEDIAN SSIM IS HIGHER FOR PACO.

metric	RMSE				SSIM			
mask	#1	#2	#3	#4	#1	#2	#3	#4
	PACO							
p. 25	2.1	4.7	2.1	4.6	0.9946	0.9696	0.9936	0.9645
median	2.6	*5.6*	2.6	5.8	0.9928	**0.9658**	0.9920	0.9584
p. 75	3.4	7.0	3.9	8.0	0.9905	0.9609	0.9896	0.9526
	Fedorov et al. [27]							
p. 25	2.6	5.1	2.5	5.2	0.9920	0.9674	0.9912	0.9575
median	3.1	*5.6*	3.1	6.4	0.9892	*0.9629*	0.9898	0.9513
p. 75	4.1	7.2	4.5	8.5	0.9870	0.9571	0.9858	0.9400
	Arias et al. [28]							
p. 25	2.9	6.6	2.8	5.9	0.9893	0.9491	0.9890	0.9508
median	3.8	7.6	3.6	7.2	0.9858	0.9470	0.9847	0.9391
p. 75	4.8	9.2	5.4	9.9	0.9833	0.9420	0.9823	0.9294

be seen indirectly in the RMSE and SSIM of the resulting reconstructed images, and approach each other slowly, which is reflected in the evolution of the Augmented Lagrangian term value (whose value is 0 at the optimum by definition of the problem). Overall, a value of $\lambda = 0.3$ gives the best results; this value is hereafter used by default.

B. Image inpainting

We have already shown our best results for the test image in Figure 3; these were obtained using patches of size 10×10 pixels and one reweighting iteration. We also conducted image inpainting experiments on all the grayscale versions of the Kodak image dataset[4] and four different masks, using patches of size 16×16 and one reweighting iteration. Figure 5 shows the masks and a few sample images of this dataset, whereas Figure 6 shows a detail of the result obtained on Kodak image #18 and mask #2. As can be observed there, the resulting image is indistinguishable from the original image in all regions but the large square to the center right. Table I shows the 0 (best) 25, 50 (median), 75 and 100 (best) quantiles of the SSIM and RMSE metrics obtained on all 24 Kodak images for each mask and compares the corresponding results of two recently published works [27], [28] which focus on the inpainting problem, with results among the best found in the literature. As can be seen, our method consistently and significantly improves upon the results of both works on all Kodak images, for all the four masks tested. Although Table I cannot be considered a thorough comparative study of inpainting methods, it provides very encouraging evidence on the competitiveness of PACO in this case.

[4]This is a dataset originally released on a CD by Kodak which was later released to the public. Many sites host a copy of this dataset. We provide our own link at `http://iie.fing.edu.uy/~nacho/data/images/kodak_color.7z` and our grayscale versions at `http://iie.fing.edu.uy/~nacho/data/images/kodak_color.7z`.

Effect of stride: Although we did not develop the matter of using strides larger than one, our implementation actually handles this case. In Figure 7 we show the inpainting quality metrics on the test image as a function of the stride (both vertical and horizontal), including $s = 1$ (the case discussed in this paper), which is the one shown in Figure 3. As the execution time is linear in the number of patches, and these are two dimensional images, a stride of size k will decrease the running time by a factor of k^2. It can be readily seen that the results are excellent even for $s = 5$ (a 50% overlap in this case since the patches are of 10×10 pixels), at a computational cost which is $1/25$th of the original one. The case $s = 10$ corresponds to the case when there is no overlapping at all, and thus the PACO constraint does not have any effect on the result.

C. Audio inpainting

In order to show the flexibility of the framework we report on a sample result on audio denoising. In this case we used the first 15 seconds of a downmixed and downsampled (from 44.1Khz to 11Khz) version of an audio track as the ground truth.[5] We then erased fragments of random length at random positions, so that we obtained an average of one erasure every 10000 samples (about one every second), each lasting on 1000 samples (0.1s) on average. The PACO inpainting algorithm was then run on windows of length 1024, a stride of $s = 16$ ($1/64$th overlap), no reweighting, $\lambda = 0.25$, and a maximum of 1000 iterations and convergence tolerance of $1e^{-8}$.

VI. CONCLUDING REMARKS AND FUTURE WORK

We have presented PACO, a simple and effective method for solving the issue of overlapping patches in patch-based signal processing problems by requiring explicitly that patches coincide at their intersections. In contrast to other works which have dealt with this issue, our method does not require weighting functions, which usually involve further assumptions on the data and the degradation model, and ad-hoc decisions. We have shown that the PACO constraint results in non-empty feasible sets. We also provided a general and simple optimization method for solving the general PACO problem which accommodates a wide array of problems and possible variants beyond patch stitching. The split formulation and the consensus constraint are naturally suited for parallel processing of very large scale signals such as astronomical images. On top of all this, we have tested the method to two classic signal processing problems, denoising and inpainting, surpassing the state of the art by a significant margin in the latter case.

Several research directions open up from here. For example, we are now working on testing PACO-based variants of already existing patch-based restoration methods. A massive parallel implementation for large scale signal processing is also under

[5]We used the original, lossless version of the composition "Se Parar" by Conrado Paulino, from the album "Quatro Climas", which can be found online in Spotify; the original track and the downsampled segment are included as supporting material and can be downloaded and reproduced by courtesy of the composer, who gave us express permission to do so.

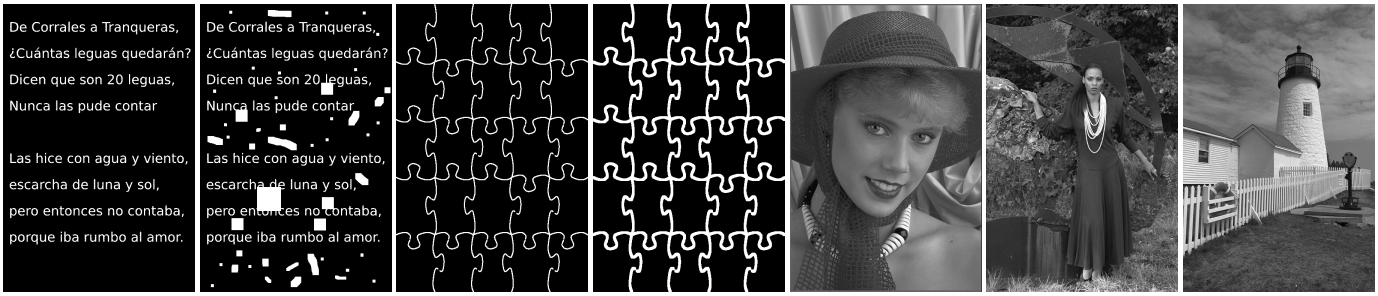

Fig. 5. Inpainting masks 1 to 4 and three sample images from the Kodak dataset. Masks 2 and 4 are more challenging due to the size of the erasures.

Fig. 6. Detail inpainting result on Kodak image #19 and mask #2. Top to bottom, left to right: mask, original, degraded, inpainted. The differences between the original and the degraded are unnoticeable with the exception of the large square to the center-right. RMSE= 7.74, SSIM=0.96315.

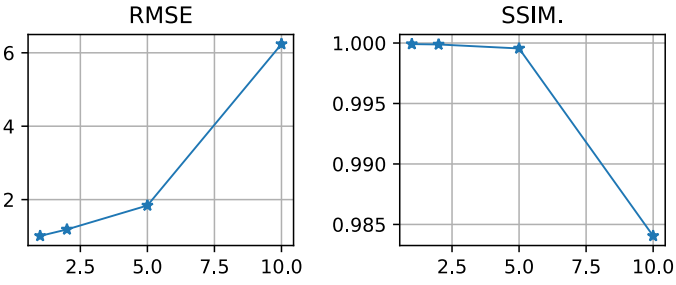

Fig. 7. Effect of stride on image inpainting performance. The case 1 corresponds to the *dense* case studied throughout this paper. The case $s = 10$ corresponds to no overlapping at all, in which case the PACO constraint does not apply. Note that the results for $s = 2$ are practically identical in all aspects to $s = 1$, and even $s = 5$ (50% overlap) gives an excellent SSIM≈ 0.999.

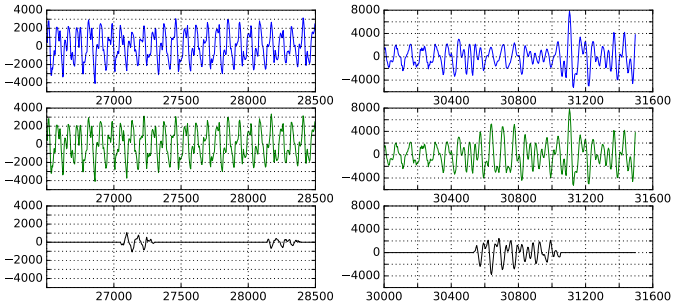

Fig. 8. Audio inpainting example. Here we show details on the inpainting of three erasures (two small on the left column and one large on the right). Each column shows, from top to bottom, the output (recovered) waveform, original waveform, and their difference; which differs from zero only where the erasures took place. The error is larger for the wider erasure (about 300 consecutive samples), than with the shorter ones (about 100 samples each); this is similar to what happens with large erasures in images. In all cases, however, the output achieves a good degree of continuity at the borders, making the result much more pleasant to listen to – the interested reader can do so by downloading the corresponding supporting material.

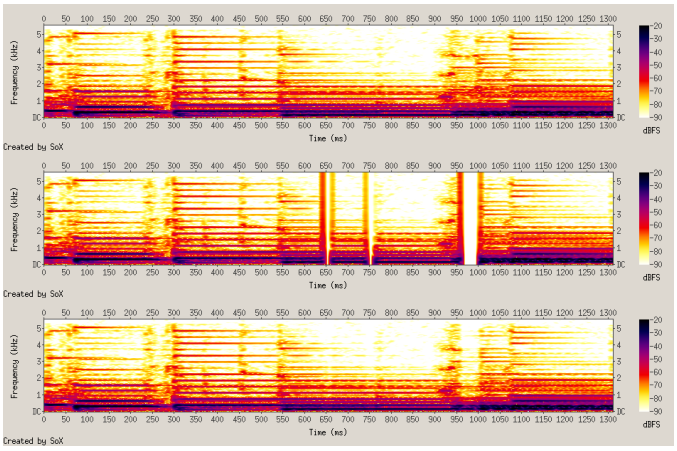

Fig. 9. Frequency analysis of the audio inpainting example. Here we show, from top to bottom, the spectrograms corresponding to the original, input (erased), and estimated output waveforms including the three erasures shown in Figure 8; these are clearly marked as white bands in the middle graph. As can be seen, the spectrogram of the recovered signal (below) is able to recover much of the low-frequency content of the signal; the high frequency harmonics are also recovered but appear fainted with respect to the original.

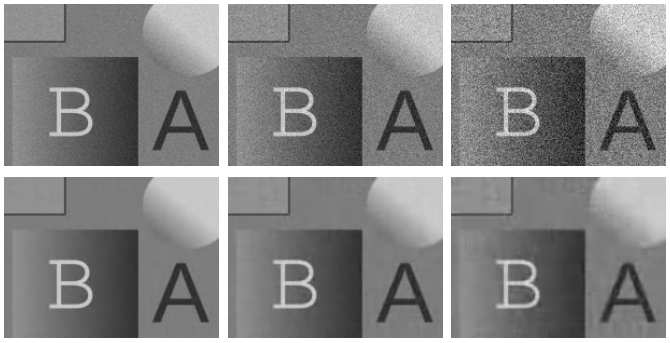

Fig. 10. Image denoising on test image. Top row: images corrupted with Gaussian noise with $\sigma = 5, 10$ and 20 respectively. Bottom row: corresponding denoised images: RMSE=$1.9, 3.2, 5.2$; SSIM=$0.9252, 0.8940, 0.8554$ The results in this case are decent, but quite behind the state of the art [29].

way. Other directions include to extend the concept of patch consensus to targets other than signal restoration. Last but not least, a deep analysis of the exceptionally good inpainting performance of PACO is required, as it is not obviously derived from or explicitly sought by our simple inpainting method.

APPENDIX

A. Proximal operator for the penalized least squares cost function

We have from (22) and $\tilde{\mathbf{V}} = \mathbf{D}\tilde{\mathbf{Y}}$ that

$$\begin{aligned}\text{prox}_{\lambda f_d}(\mathbf{B}) = \quad & \arg\min_{\mathbf{A}} \Big\{ \textstyle\sum_{ij} w_{ij}|a_{ij}| + \\ & \tfrac{1}{2\tau}\|\mathbf{A}-\tilde{\mathbf{V}}\|_F^2 + \tfrac{1}{2\lambda}\|\mathbf{A}-\mathbf{B}\|_F^2 \Big\}.\end{aligned}$$

Since (27) is separable in the elements of $\mathbf{A}$, we can reduce the problem to solving the following scalar proximal operator,

$$\begin{aligned}\text{prox}_{\lambda f_d}(b_{ij}) = \quad & \arg\min_a \{ w_{ij}|a| + \\ & \tfrac{1}{2\tau}(a-\tilde{v}_{ij}) + \tfrac{1}{2\lambda}(a-b_{ij}) \}.\end{aligned}$$

Let $\partial|a|$ denote the sub-differential of the absolute value function, $\partial|a| = \text{sgn}(a), a \neq 0$ and $\partial|0| = [-1, 1]$. The optimality condition is obtained after differentiating (27),

$$0 \in w_{ij}\partial|a_{ij}| + (1/\tau)(a_{ij}-\tilde{v}_{ij}) + (1/\lambda)(a_{ij}-b_{ij}). \quad (27)$$

We have that $a_{ij} = 0$ whenever

$$\begin{aligned} 0 &\in w_{ij}[-1,1] - (1/\tau)\tilde{v}_{ij} - (1/\lambda)b_{ij} \\ (1/\tau)\tilde{v}_{ij} + (1/\lambda)b_{ij} &\in [-w_{ij}, w_{ij}] \\ |\lambda\tilde{v}_{ij} + \tau b_{ij}| &\leq \lambda\tau w_{ij} \end{aligned} \quad (28)$$

For $a_{ij} > 0$ to happen we need that

$$\begin{aligned} 0 &= w_{ij} + (1/\tau)(a_{ij}-\tilde{v}_{ij}) + (1/\lambda)(a_{ij}-b_{ij}) \\ 0 &= \tau\lambda w_{ij} + \lambda(a_{ij}-\tilde{v}_{ij}) + \tau(a_{ij}-b_{ij}) \\ (\lambda+\tau)a_{ij} &= \lambda\tilde{v}_{ij} + \tau b_{ij} - \lambda\tau w_{ij}. \end{aligned} \quad (29)$$

Analogously for $a_{ij} < 0$ we arrive at

$$(\lambda+\tau)a_{ij} = \lambda\tilde{v}_{ij} + \tau b_{ij} + \lambda\tau w_{ij}. \quad (30)$$

Dividing (28)–(30) by $(\lambda+\tau)$ and defining $\theta_{ij} = \frac{\lambda\tau}{\lambda+\tau}w_{ij}$ we arrive at

$$a_{ij} = \mathcal{T}_{\theta_{ij}}\left(\frac{\lambda}{\lambda+\tau}\tilde{v}_{ij} + \frac{\tau}{\lambda+\tau}b_{ij}\right).$$

REFERENCES

[1] J. A. Moorer, "A note on the implementation of audio processing by short-term fourier transform," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, vol. 2017-October, 2017, pp. 156–159. [Online]. Available: www.scopus.com

[2] V. Gnann and J. Becker, "Signal reconstruction from multiresolution stft magnitudes with mutual initialization," in *Proceedings of the AES International Conference*, 2012, pp. 274–279. [Online]. Available: www.scopus.com

[3] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude stft spectrogram based on spectrogram consistency," in *13th International Conference on Digital Audio Effects, DAFx 2010 Proceedings*, 2010, cited By :29. [Online]. Available: www.scopus.com

[4] M. M. Goodwin, "Realization of arbitrary filters in the stft domain," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 353–356, cited By :1. [Online]. Available: www.scopus.com

[5] Z. Liu, L. Chen, and J. Yang, "An image reconstruction algorithm using patch-based locally optimal wiener filtering," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 36, no. 11, pp. 2556–2562, 2014, cited By :3. [Online]. Available: www.scopus.com

[6] I. denoising by sparse 3-D transform-domain collaborative filtering, "K. dabov and a. foi and v. katkovnik and k. egiazarian," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, 2006.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008, cited By :930. [Online]. Available: www.scopus.com

[9] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008, cited By :285. [Online]. Available: www.scopus.com

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010, cited By :1302. [Online]. Available: www.scopus.com

[11] O. G. Guleryuz, "Weighted averaging for denoising with overcomplete dictionaries," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 3020–3034, Dec 2007.

[12] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[13] V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 249–261, 2016, cited By :39. [Online]. Available: www.scopus.com

[14] M. H. Alkinani and M. R. El-Sakka, "Patch-based models and algorithms for image denoising: a comparative review between patch-based images denoising methods for additive noise reduction," *Eurasip Journal on Image and Video Processing*, vol. 2017, no. 1, 2017. [Online]. Available: www.scopus.com

[15] Z. Dengwen and S. Xiaoliu, "Image denoising using weighted averaging," in *2009 WRI International Conference on Communications and Mobile Computing*, vol. 1, Jan 2009, pp. 400–403.

[16] O. G. Sezer and Y. Altunbasak, "Weighted average denoising with sparse orthonormal transforms," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 3849–3852.

[17] B. Wang, T. Lu, and Z. Xiong, "Adaptive boosting for image denoising: Beyond low-rank representation and sparse coding," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 1400–1405.

[18] J. Feng, L. Song, X. Huo, X. Yang, and W. Zhang, "An optimized pixel-wise weighting approach for patch-based image denoising," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 115–119, Jan 2015.

[19] D. Carrera, G. Boracchi, A. Foi, and B. Wohlberg, "Sparse overcomplete denoising: Aggregation versus global optimization," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1468–1472, Oct 2017.

[20] J. Eckstein and D. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[21] N. Parikh and S. P. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: https://doi.org/10.1561/2400000003

[22] M. Aharon, M. Elad, and A. Bruckstein, "k -svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[23] J. Boyle and R. Dykstra, "A method for finding projections onto the intersection of convex sets in hilbert spaces," in *Advances in Order Restricted Statistical Inference*, ser. Lecture Notes in Statistics, T. F. W. R. Dykstra, T. Robertson, Ed. Springer, New York, NY, 1986, vol. 37.

[24] E. Esser, X. Zhang, and T. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.

[25] I. Ramirez and G. Sapiro, "Universal regularizers for robust sparse coding and modeling," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3850–3864, Sept 2012.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 1, Jan. 2004.

[27] V. Fedorov, G. Facciolo, and P. Arias, "Variational Framework for Non-Local Inpainting," *Image Processing On Line*, vol. 5, pp. 362–386, 2015.

[28] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez, "Non-Local Patch-Based Image Inpainting," *Image Processing On Line*, vol. 7, pp. 373–385, 2017.

[29] N. Pierazzo, J.-M. Morel, and G. Facciolo, "Multi-scale DCT denoising," *Image Processing On Line*, vol. 7, pp. 288–308, 2017.