Constrained and Unconstrained Inverse Potts Modeling for Joint Image Super-Resolution and Segmentation

Dario Mylonopoulos[1], Pasquale Cascarano[2], Luca Calatroni[3], Elena Loli Piccolomini[4]

[1]Department of Computer Science and Engineering, University of Bologna
(`dario.mylonopoulos@studio.unibo.it`)
[2]Department of Mathematics, University of Bologna (`pasquale.cascarano2@unibo.it`)
[3]CNRS, Universite Côte dAzur, Inria, Sophia-Antipolis, France (`calatroni@i3s.unice.fr`)
[4]Department of Computer Science and Engineering, University of Bologna (`elena.loli@unibo.it`)

Communicated by Gregory Randall   *Demo edited by* Pascal Monasse

Abstract

In this work we consider two methods for joint single-image super-resolution and image partitioning. The proposed approaches rely on a constrained and on an unconstrained version of the inverse Potts model where an ℓ_0 regularization prior on the image gradient is used for promoting piecewise constant solutions. For the numerical solution of both models, we provide a unified implementation based on the Alternating Direction Method of Multipliers (ADMM). Upon suitable assumptions on both model operators and on the algorithmic parameters involved, we show that all the ADMM subproblems admit closed-form solutions, thus making the resulting algorithms computationally very cheap even when high-dimensional data are considered. Numerical details of the implementation of both models are given and several experiments are carried out on both synthetic and natural images to underline the accuracy and the computational efficiency of the models.

Source Code

The Matlab source code, the code online documentation and the online demo are available from the web page of this article[1]. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: single-image super-resolution; image segmentation; inverse Potts model; ADMM

---

[1]https://doi.org/10.5201/ipol.2022.393

1   Introduction

Single-Image Super-Resolution (SISR) and Image Partitioning (IP) are two popular tasks in the field of image processing. In SISR, the objective is to enhance the spatial resolution of a given Low-Resolution (LR) and possibly blurred and noisy image so as to retrieve a High-Resolution (HR) version enhancing the quality of the LR data. In IP (a technique which is often referred to as image segmentation), the objective is to extract from a given digital image regions of interest on the basis of geometric/semantic information. Such task is typically performed to facilitate subsequent data classification and labeling. A standard approach to extract a suitable IP from LR data consists in performing the SISR and the IP steps in a disjoint sequential manner. A clear limitation of such sequential strategy is that the quality of the partitioning obtained as final result depends on the quality of the super-resolved image obtained after performing the former reconstruction step. Typically, this depends on the SISR model used and on the accurate choice of its hyperparameters. To overcome these limitations, a joint SR and IP approach performing both tasks at the same time has been proposed, e.g., in [10, 6, 3] based on a Bayesian approach.

In this paper, we consider a joint SISR and IP model assuming, for a given vectorized image $\mathbf{g} \in \mathbb{R}^{N_l}$, the following image formation model

$$\mathbf{g} = \mathbf{SHu} + \boldsymbol{\eta}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{N_h}$ is the unknown HR image defined on a finer grid of dimension N_h so that $N_h = L^2 N_l, L \in \mathbb{N}, L \geq 1$, $\mathbf{H} \in \mathbb{R}^{N_h \times N_h}$ is the operator corresponding to the modeling of a Gaussian space-invariant blur with standard deviation $\sigma_{\mathbf{H}} > 0$, $\mathbf{S} \in \mathbb{R}^{N_l \times N_h}$ is the discrete down-sampling operator and $\boldsymbol{\eta} \in \mathbb{R}^{N_l}$ denotes the realisation of a Gaussian random variable of mean 0 and standard deviation equal to $\sigma_{\boldsymbol{\eta}} > 0$.

Finding $\mathbf{u}$ given $\mathbf{g}$ as in (1) is an ill-posed (under-determined) inverse problem whose solution requires the use of suitable regularization. In a variety of works [12, 13, 11, 4], Storath et al. considered the inverse Potts regularization model for joint image restoration and segmentation. Heuristically, such approach is based on the use of a penalized regularization functional defined in terms of an ℓ_0-type gradient smoothing prior reducing noise and preserving distinctive details (such as image edges). Furthermore, due to its strong sparsification properties, it favors simplified and almost-partitioned reconstructions that can be easily used for subsequent segmentation purposes. Inspired by this work, in [2] the authors extended the aforementioned approach to the problem of joint segmentation and SISR. To show the effectiveness of such modeling, as an example, in Figure 1 some results are reported. Namely, for given unknown piecewise constant (Figure 1 (a1)) and natural (Figure 1 (d1)) HR image data, the super-resolved results obtained by means of the approach in [2] are reported in Figure 1 (c1) and Figure 1 (f1), respectively. As it can be clearly seen, the method provides an accurate partitioning of both images which can be a precious pre-processing for further tasks. To motivate this further, we report in Figure 1 (c2), in Figure 1 (f2) and in Figure 2 (d) an example of how a good joint SR and IP process helps in the detection of the objects of interest, which are there shown in terms of a given binary mask computed through standard segmentation algorithms or super-imposed to the image, respectively.

As far as the numerical solution of the inverse Potts joint SISR and IP model is considered, in [2] an Alternating Direction Method of Multipliers (ADMM) algorithm has been applied and fixed-point convergence results were shown. From a computational point of view, the most expensive step in the proposed algorithm was shown to be the Conjugate-Gradient (CG) type solver required to solve the linear system arising in the quadratic substeps of the ADMM scheme. In the case of large-scale data, this can of course be a computational burden preventing the model to be used in practice. Under specific structural assumptions on the down-sampling operator $\mathbf{S}$, however, quadratic substeps can be solved in closed-form by using Fourier-based techniques combined with the application of

DARIO MYLONOPOULOS, PASQUALE CASCARANO, LUCA CALATRONI, ELENA LOLI PICCOLOMINI

Woodbury's formula [18]. This technique significantly improves the computational complexity in several applications such as 3D imaging [14] and automatic parameter selection strategies [9].

An alternative approach providing a joint super-resolved and partitioned version of the image but relying on a constrained approach still defined in terms of an ℓ_0-type constraint on the gradient was proposed in [8]. This alternative approach can be used in place of the unconstrained one whenever information on the number of jumps (i.e. image discontinuities) of the desired solution is available.
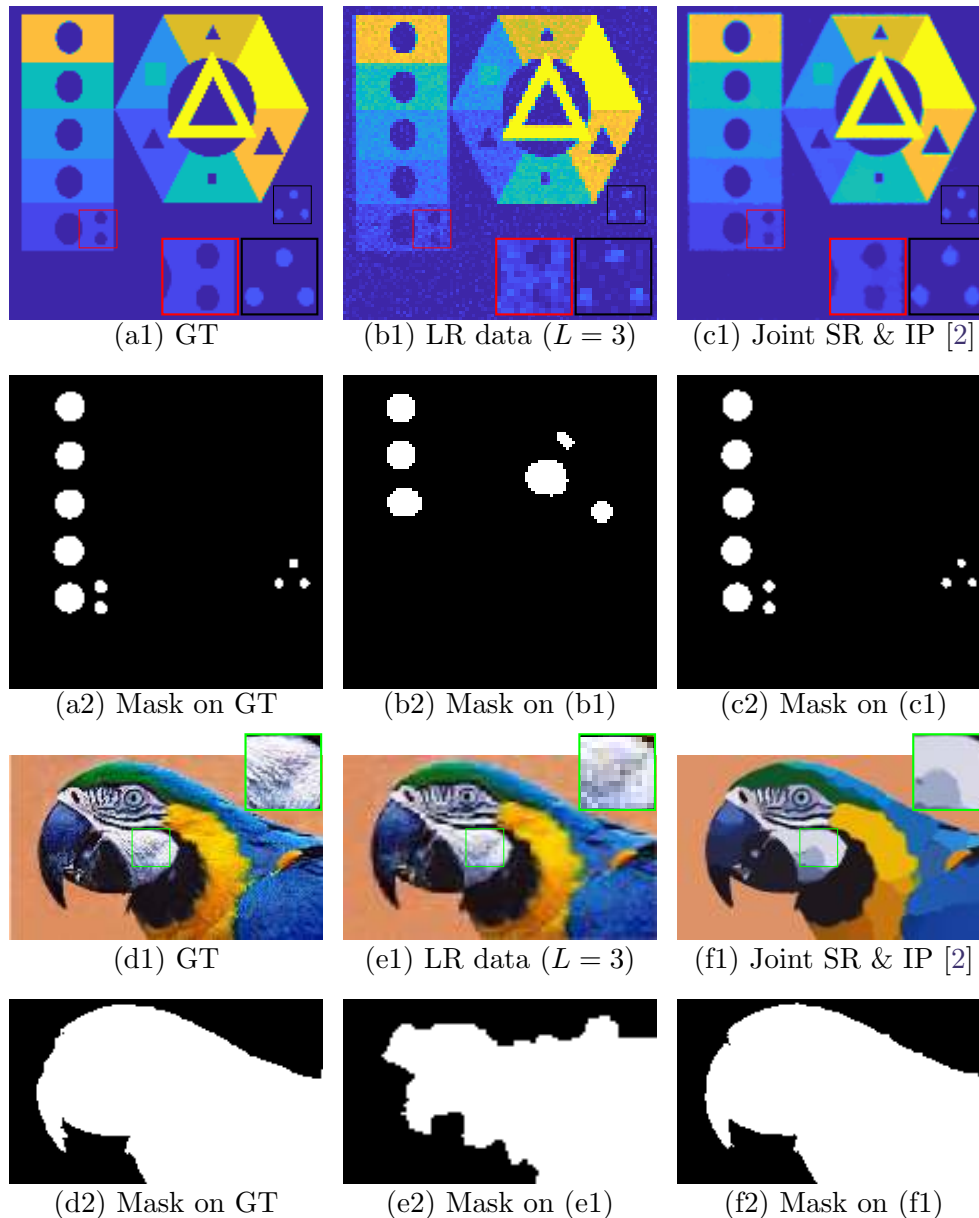

Figure 1: **Joint SISR and IP of a piecewise constant (upper rows) and natural image (lower rows)**. The simulated LR acquisitions are corrupted according to model (1) by setting $L = 3$, $\sigma_\eta = 0.01$, $\sigma_H = 1.0$. The SR images have been obtained by solving the proposed jump-penalizing model (4) as in [2] with $\mu = 0.0024$ and $\mu = 0.023$, respectively.

Contribution We consider a constrained and an unconstrained approach based on the ℓ_0 smoothing of the image gradient to jointly perform SISR and IP. The choice of the regularization considered is justified as an effective way of simplifying as much as possible image contents in order to favor IP. For both models, an ADMM iterative scheme is used to compute efficiently the desired numerical solution.

The unconstrained approach has been previously presented in [2], the only difference being the
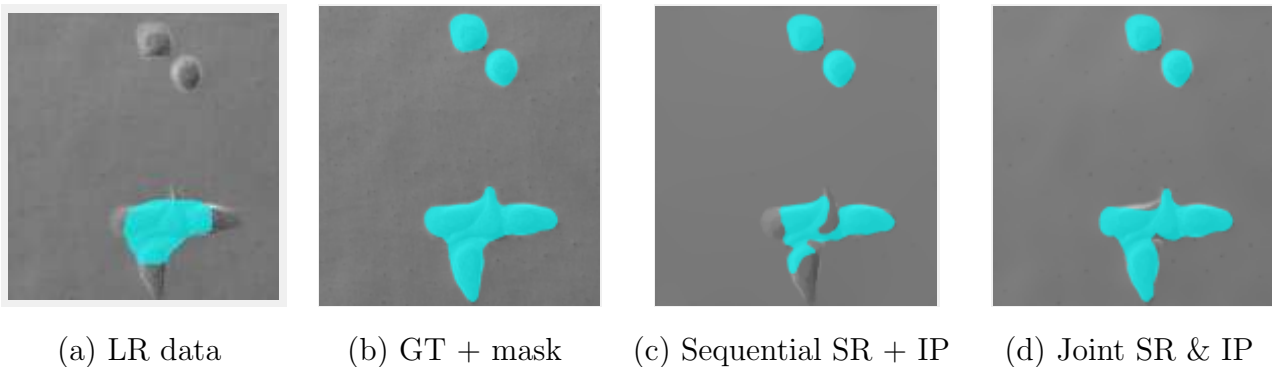
| (a) LR data | (b) GT + mask | (c) Sequential SR + IP | (d) Joint SR & IP |

Figure 2: **Cell detection on low- and super-resolved data.** Masks are computed on the cells of interest are coloured cyan. (a) LR image ($L = 5$). (b) GT image with superimposed mask. (c) SR image computed by first applying the bicubic interpolation algorithm to the LR data and then the IP algorithm based on constrained gradient smoothing described in [17]. (d) SR image obtained from LR data by solving (4) using the proposed approach, see [2].

use of a new and much faster solver for the linear system arising in the quadratic step of the ADMM scheme. Upon suitable assumptions on the downsampling operator considered, such step can be solved in closed-form after a clever application of Woodbury's formula [18] which, after some manipulations, allows the use of Fourier-based techniques significantly improving the computational complexity, see, e.g., [14, 9], thus making the overall iterative scheme fast and suitable also to the analysis of large scale data with large SR factors.

Analogously, we extend to the SISR problem the constrained ℓ_0-based formulation proposed in [8] for image denoising only. Also in this case, we consider a similar efficient ADMM algorithm for solving the constrained model and report several numerical comparisons between the two approaches considered for different levels of blur and noise in the data.

2   Joint SISR and IP Via Inverse Potts Modeling

We introduce in this section the main notation and definitions used in the following and we present both the constrained and the unconstrained ℓ_0-gradient based models assumed to perform joint SR and IP.

A discrete jump penaliser. For $N \in \mathbb{N}$, let $\Gamma = \{\Gamma_1, \ldots, \Gamma_K\}$ be a partition of the set $\{1, \ldots, N\}$, that is a set of $1 \leq K \leq N$ non-empty subsets Γ_i such that for all $\Gamma_i \in \Gamma$:

- $\Gamma_i \subset \{1, \ldots, N\}$, for $i = 1 \ldots K$,

- $\Gamma_i \cap \Gamma_j = \emptyset$, for $i, j = 1 \ldots K$, with $i \neq j$,

- $\cup_{i=1}^{K} \Gamma_i = \{1, \ldots, N\}$.

Given a generic $\mathbf{x} \in \mathbb{R}^N$ and an element $\Gamma_i \in \Gamma$, we denote by $\mathbf{x}_{\Gamma_i}$ the subvector extracted from $\mathbf{x}$ whose entries are specified by the indexes in Γ_i. We can then define the ℓ_0 pseudo-norm with respect to the partition Γ as the function $\|\cdot\|_0^\Gamma : \mathbb{R}^N \to \mathbb{R}_+$ as

$$\|\mathbf{x}\|_0^\Gamma := \sum_{i=1}^{K} |\|\mathbf{x}_{\Gamma_i}\||_0, \tag{2}$$

where $|\cdot|_0$ denotes the non-zero counting scalar function which is equal to 0 whenever the argument is zero and one otherwise and $\|\cdot\|$ stands for any ℓ_p norm, $p \geq 1$. As a particular instance of (2),

if $K = N$ and $\Gamma_i = \{i\}$ for all $i = 1 \ldots N$, then the ℓ_0 pseudo-norm function counts the number of non-zero entries of the vector $\mathbf{x} \in \mathbb{R}^N$.

By choosing the standard Euclidead norm (i.e. $p = 2$) in (2) and denoting by $\mathbf{D} \in \mathbb{R}^{2N_h \times N_h}$ the discrete finite difference operator defined as the block matrix $(\mathbf{D}_x; \mathbf{D}_y)$, $\mathbf{D}_x, \mathbf{D}_y \in \mathbb{R}^{N_h \times N_h}$, being the first order difference operator along the horizontal and vertical axes, respectively, we define a function $R_{\ell_0} : \mathbb{R}^{N_h} \to \mathbb{R}_+$ as

$$R_{\ell_0}(\mathbf{u}) = \|\mathbf{D}\mathbf{u}\|_0^\Gamma, \tag{3}$$

where $\Gamma_i = \{i, i + N_h\}$, i.e. the set containing the indices corresponding to the vertical and horizontal differences at the i-th pixel, for $i = 1 \ldots N_h$. The function R_{ℓ_0} in (3) penalizes the number of jumps in $\mathbf{u}$ in terms of the non-zero values of its gradient magnitudes, jointly accounted for each pixel and thus favoring image partitioning in sharp piecewise constant patches.

To improve the readability of the following section, by Γ we will always refer to the partition such that $\Gamma_i = \{i, i + N_h\}$ for $i = 1 \ldots N_h$. Therefore given $\mathbf{x} \in \mathbb{R}^{2N_h}$ we will denote by $\mathbf{x}_{\Gamma_i}$ the subvector of $\mathbf{x}$ whose entries are $(\mathbf{x}_i, \mathbf{x}_{i+N_h})$ for $i = 1 \ldots N_h$. Finally, we will omit Γ when using the notation referring to the function defined in (2).

The ℓ_2-R_{ℓ_0} unconstrained/constrained models   For solving (1) we introduce two R_{ℓ_0}-regularized SISR models, both in an unconstrained and in a constrained fashion. Due to the non-convexity of the regularization under consideration, we remark that such formulations are indeed not equivalent, hence they deserve a separate discussion.

The unconstrained inverse Potts model has previously been considered in [2] for the problem of SISR. It computes solutions $\mathbf{u}^*$ as

$$\mathbf{u}^* \in \arg\min_{\mathbf{u} \in \mathbb{R}^{N_h}} \frac{1}{2} \|\mathbf{S}\mathbf{H}\mathbf{u} - \mathbf{g}\|_2^2 + \mu \|\mathbf{D}\mathbf{u}\|_0, \tag{4}$$

where the ℓ_2 fidelity term describes the presence of additive white Gaussian noise statistics and $\mu > 0$ denotes the regularization parameter.

The form of the analogous constrained model is inspired by the formulation proposed in [8] for simple image denoising problems (i.e. with no forward operator) and adapted here to the SISR task. It computes solutions $\mathbf{u}^*$ as

$$\mathbf{u}^* \in \arg\min_{\mathbf{u} \in \mathbb{R}^{N_h}} \frac{1}{2} \|\mathbf{S}\mathbf{H}\mathbf{u} - \mathbf{g}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{D}\mathbf{u}\|_0 \leq \alpha \in \mathbb{N}. \tag{5}$$

The parameter α can be interpreted as the number of expected jumps in the desired solution. Choosing a proper value of α in (5) may be more practical than choosing μ in (4) whenever edge-maps specifying the number of edges for adjusting the flatness of the output are available. On the other hand, the choice of μ may be driven by standard a posteriori parameter rules, such as, e.g., the Morozov discrepancy principle [7] or whiteness-based principles [9].

Anyway, both models (4) and (5) favor ℓ_0-type gradient smoothing, thus favoring simplified, piecewise-constant solutions which are amenable for precise IP.

We finally remark the connection of both models (4) and (5) with the piecewise constant Mumford-Shah model, as analyzed, for example in [5].

3   ADMM Optimization

To numerically solve problems (4) and (5), we propose to use the iterative Alternating Direction Method of Multipliers (ADMM), a common optimization strategy which has been largely studied

both in convex [1] and in non-convex [15] regimes. For both the unconstrained and the constrained problem, the ADMM iterations are defined in terms of a suitable variable splitting corresponding to the solution of three different sub-problems which can be efficiently solved either by standard solvers or via some manipulations, as described in the following.

3.1 Variable Splitting: the Unconstrained Case

Following [2], the optimization problem in (4) can be equivalently reformulated in terms of an auxiliary variable $\mathbf{z} \in \mathbb{R}^{2N_h}$ defined by $\mathbf{z} := \mathbf{D}\mathbf{u}$ as

$$\underset{\mathbf{u} \in \mathbb{R}^{N_h},\ \mathbf{z} \in \mathbb{R}^{2N_h}}{\arg\min}\quad \frac{1}{2}\|\mathbf{SHu} - \mathbf{g}\|_2^2 \ + \ \mu\|\mathbf{z}\|_0, \tag{6}$$
$$\text{s.t.}\quad \mathbf{z} = \mathbf{D}\mathbf{u}.$$

In particular, for a given initialization $\mathbf{u}^0 \in \mathbb{R}^{N_h}$ and $\mathbf{z}^0 \in \mathbb{R}^{2N_h}$, under suitable assumptions on the gradient operator $\mathbf{D}$, a fixed-point convergence result for the sequence $\{\mathbf{u}_k\}_{k\in\mathbb{N}}$ generated, for initial $\mathbf{u}^0 \in \mathbb{R}^{N_h}$ and $\mathbf{z}^0 \in \mathbb{R}^{2N_h}$, by the iteration

$$\begin{cases} \mathbf{z}^{k+1} & \in \underset{\mathbf{z}\in\mathbb{R}^{2N_h}}{\arg\min}\ \mu\|\mathbf{z}\|_0 + \dfrac{\beta^k}{2}\|\mathbf{z} - (\mathbf{D}\mathbf{u}^k + \dfrac{\boldsymbol{\lambda}^k}{\beta^k})\|_2^2, \\[2mm] \mathbf{u}^{k+1} & = \underset{\mathbf{u}\in\mathbb{R}^{N_h}}{\arg\min}\ \dfrac{1}{2}\|\mathbf{SHu} - \mathbf{g}\|_2^2 + \dfrac{\beta^k}{2}\|\mathbf{D}\mathbf{u} - (\mathbf{z}^{k+1} - \dfrac{\boldsymbol{\lambda}^k}{\beta^k})\|_2^2, \\[2mm] \boldsymbol{\lambda}^{k+1} & = \boldsymbol{\lambda}^k - \beta^k(\mathbf{z}^{k+1} - \mathbf{D}\mathbf{u}^{k+1}), \end{cases} \tag{U-SISR}$$

is given for an increasing sequence of penalty parameters $\{\beta^k\}_{k\in\mathbb{N}}$ such that $\beta^k = k(1+\epsilon), \epsilon > 0$ for each k, see [2, Theorems 2-3].

3.2 Variable Splitting: the Constrained Case

Similarly, we can reformulate the constrained optimization problem (5) as

$$\underset{\mathbf{u} \in \mathbb{R}^{N_h},\ \mathbf{z} \in \mathbb{R}^{2N_h}}{\arg\min}\quad \frac{1}{2}\|\mathbf{SHu} - \mathbf{g}\|_2^2 \ + \ i_{\{\|\cdot\|_0\leq\alpha\}}(\mathbf{z}), \tag{7}$$
$$\text{s.t.}\quad \mathbf{z} := \mathbf{D}\mathbf{u},$$

where $i_{\{\|\cdot\|_0\leq\alpha\}}(\cdot) : \mathbb{R}^{2N_h} \to \{0, +\infty\}$ denotes the indicator function of the non-convex set $\{\mathbf{z} \in \mathbb{R}^{2N_h} : \|\mathbf{z}\|_0 \leq \alpha\}$. By following [8] and considering a sequence of increasing penalty parameters $\{\beta^k\}_{k\in\mathbb{N}}$ for guaranteeing convergence, we thus seek for an approximation of an optimal solution of (7) by iterating the following scheme for initial $\mathbf{u}^0 \in \mathbb{R}^{N_h}$ and $\mathbf{z}^0 \in \mathbb{R}^{2N_h}$

$$\begin{cases} \mathbf{z}^{k+1} & \in \underset{\mathbf{z}\in\mathbb{R}^{2N_h}}{\arg\min}\ i_{\{\|\cdot\|_0\leq\alpha\}}(\mathbf{z}) + \dfrac{\beta^k}{2}\|\mathbf{z} - (\mathbf{D}\mathbf{u}^k + \dfrac{\boldsymbol{\lambda}^k}{\beta^k})\|_2^2, \\[2mm] \mathbf{u}^{k+1} & = \underset{\mathbf{u}\in\mathbb{R}^{N_h}}{\arg\min}\ \dfrac{1}{2}\|\mathbf{SHu} - \mathbf{g}\|_2^2 + \dfrac{\beta^k}{2}\|\mathbf{D}\mathbf{u} - (\mathbf{z}^{k+1} - \dfrac{\boldsymbol{\lambda}^k}{\beta^k})\|_2^2, \\[2mm] \boldsymbol{\lambda}^{k+1} & = \boldsymbol{\lambda}^k - \beta^k(\mathbf{z}^{k+1} - \mathbf{D}\mathbf{u}^{k+1}). \end{cases} \tag{C-SISR}$$

We will now provide more details on how to solve the different substeps for both ADMM schemes (U-SISR) and (C-SISR).

3.3 Solving the ℓ_0 Sub-Steps

Due to the structure of the ℓ_0 term defined in (2), the objective function corresponding to the **z**-subproblem in (U-SISR) is separable. Hence, denoting by $\mathbf{v}^k$ the vector $\mathbf{D}\mathbf{u}^k + \frac{\boldsymbol{\lambda}^k}{\beta^k}$ a solution $\mathbf{z}^{k+1} \in \mathbb{R}^{2N_h}$ can thus be computed by solving N_h 2D-optimization problems of the form

$$\arg\min_{\mathbf{x}\in\mathbb{R}^2} \delta^k \|\mathbf{x}\|_0 + \|\mathbf{x} - \mathbf{v}^k_{\Gamma_i}\|_2^2, \quad i = 1\ldots N_h, \tag{8}$$

where $\delta^k = \frac{2\mu}{\beta^k}$. To solve (8), we apply the 2D hard-thresholding operator HT_{δ^k} to $\mathbf{v}^k$ as in [17]. Then, for $i = 1\ldots N_h$

$$\mathbf{z}^{k+1}_{\Gamma_i} = \mathrm{HT}_{\delta^k}(\mathbf{v}^k_{\Gamma_i}) = \begin{cases} 0 & \text{if } \|\mathbf{v}^k_{\Gamma_i}\|_2^2 < 2\delta^k, \\ \mathbf{v}^k_{\Gamma_i}, & \text{if } \|\mathbf{v}^k_{\Gamma_i}\|_2^2 \geq 2\delta^k. \end{cases}$$

The steps for the solution of the ℓ_0 subproblem in (U-SISR) are summarized in Algorithm 1.

Algorithm 1 – ℓ_0 sub-step for (U-SISR)

input: $\mathbf{D}\mathbf{u}^k \in \mathbb{R}^{2N_h}$, $\lambda^k > 0$, $\beta^k > 0$, $\mu > 0$
output: $\mathbf{z}^{k+1}$
1: $\mathbf{v}^k \leftarrow \mathbf{D}\mathbf{u}^k + \frac{\lambda^k}{\beta^k}$
2: **for** $i = 1\ldots N_h$ **do**
3:     $\mathbf{z}^{k+1}_{\Gamma_i} \leftarrow \begin{cases} 0 & \text{if } \|\mathbf{v}^k_{\Gamma_i}\|_2^2 < \frac{2\mu}{\beta^k} \\ \mathbf{v}^k_{\Gamma_i} & \text{otherwise} \end{cases}$
4: **end for**

Concerning the constrained algorithm (C-SISR), a solution of the corresponding ℓ_0 substep associated to **z** can be computed by projecting $\mathbf{v}^k$ by following [8, Proposition 1]

$$\mathbf{z}^{k+1} = \begin{cases} \mathbf{v}^k, & \text{if } \|\mathbf{v}^k\|_0 \leq \alpha, \\ \tilde{\mathbf{v}}^k & \text{otherwise,} \end{cases}, \quad \text{with} \quad \tilde{\mathbf{v}}^k_{\Gamma_i} := \begin{cases} \mathbf{v}^k_{\Gamma_i} & i \in \{(1),\ldots,(\alpha)\} \\ 0 & i \in \{(\alpha+1),\ldots,(N_h)\} \end{cases},$$

and the indexes $(1),\ldots(N_h)$ are computed by sorting in descending order the ℓ_2 norms of the subvectors $\mathbf{v}^k_{\Gamma_i}$ for $i = 1\ldots N_h$, and relabelling them accordingly. In other words, $\mathbf{z}^{k+1}$ is computed by replacing by zero the $N_h - \alpha$ subvectors $\mathbf{v}^k_{\Gamma_i}$ of $\mathbf{v}^k$ with the smallest ℓ_2 norm.

The steps for the solution of the ℓ_0 subproblem in (C-SISR) case are summarized in Algorithm 2.

Algorithm 2 – ℓ_0 sub-step for (C-SISR)

input: $\mathbf{D}\mathbf{u}^k \in \mathbb{R}^{2N_h}$, $\lambda^k > 0$, $\beta^k > 0$, $\alpha \in \mathbb{N}$
output: $\mathbf{z}^{k+1}$
1: $\mathbf{v}^k \leftarrow \mathbf{D}\mathbf{u}^k + \frac{\lambda^k}{\beta^k}$
2: Compute $\{(1),\ldots,(N_h)\}$ by sorting the subvectors of $\mathbf{v}^k$ in descending order in terms of their ℓ_2 norm and compute $\tilde{\mathbf{v}}^k$.
3: Compute:

$$\mathbf{z}^{k+1} = \begin{cases} \mathbf{v}^k, & \text{if } \|\mathbf{v}^k\|_0 \leq \alpha \\ \tilde{\mathbf{v}}^k & \text{otherwise.} \end{cases}$$

3.4 Solving the ℓ_2-ℓ_2 Sub-Steps: Exploiting the Structure of H, S and D

On the structure of H. Due to the space-invariant assumption on the blur kernel, the matrix $\mathbf{H} \in \mathbb{R}^{N_h \times N_h}$ is a 2D convolution matrix. Assuming periodic boundary conditions, $\mathbf{H}$ thus takes the form of a block circulant matrix with circulant blocks (BCCB), hence it can be easily diagonalized by the 2D discrete Fourier Transform, whose unitary matrix will be denoted by $\mathbf{F} \in \mathbb{R}^{N_h \times N_h}$, as

$$\mathbf{H} = \mathbf{F}^H \mathbf{\Lambda} \mathbf{F} \qquad \text{with} \qquad \mathbf{F}^H \mathbf{F} = \mathbf{F} \mathbf{F}^H = \mathbf{I}_{N_h}, \tag{9}$$

and where $\mathbf{\Lambda} \in \mathbb{R}^{N_h \times N_h}$ is diagonal. The computation of matrix-vector products of the form $\mathbf{Hz}$ or $\mathbf{H}^H \mathbf{z}$ can thus be performed element-wise in the Fourier domain.

On the structure of S. We consider a down-sampling matrix $\mathbf{S} \in \mathbb{R}^{N_l \times N_h}$ in the form of a decimation operator removing selected rows and columns from the vectorized image it is applied to, see Figure 3 for a schematic example of the action of $\mathbf{S}$ for $N_h = 16$ and $L = 2$. Following [18], we assume in particular that the operator $\mathbf{S}^H \in \mathbb{R}^{N_h \times N_l}$ interpolates the decimated image with zeros. The matrix $\mathbf{S}$ is thus unstructured and, in particular, it cannot be diagonalized by the 2D discrete Fourier Transform. As a consequence, the computation of matrix-vector products of the form $\mathbf{Sz}$ and $\mathbf{S}^H \mathbf{w}$ for $\mathbf{z} \in \mathbb{R}^{N_h}$ and $\mathbf{w} \in \mathbb{R}^{N_l}$ cannot be a priori fastly computed. However, following [18] some considerations can still be done to improve the computational efficiency. For an integer d, denoting by $\mathbf{J}_d \in \mathbb{R}^{d \times d}$ a matrix of all ones, by $\mathbf{1}_d$ the d-dimensional vector of ones and by $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ the identity matrix, the following chain of identities holds

$$\mathbf{F}\mathbf{S}^H\mathbf{S}\mathbf{F}^H := \frac{1}{L^2} \left(\mathbf{J}_L \otimes \mathbf{I}_{n_r} \right) \otimes \left(\mathbf{J}_L \otimes \mathbf{I}_{n_c} \right) \tag{10}$$

$$= \frac{1}{L^2} \left(\mathbf{1}_L \mathbf{1}_L^T \otimes \mathbf{I}_{n_r} \mathbf{I}_{n_r} \right) \otimes \left(\mathbf{1}_L \mathbf{1}_L^T \otimes \mathbf{I}_{n_c} \mathbf{I}_{n_c} \right) \tag{11}$$

$$= \frac{1}{L^2} \left(\left(\mathbf{1}_L \otimes \mathbf{I}_{n_r} \right) \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_r} \right) \right) \otimes \left(\left(\mathbf{1}_L \otimes \mathbf{I}_{n_c} \right) \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_c} \right) \right) \tag{12}$$

$$= \frac{1}{L^2} \left(\left(\mathbf{1}_L \otimes \mathbf{I}_{n_r} \right) \otimes \left(\mathbf{1}_L \otimes \mathbf{I}_{n_c} \right) \right) \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_r} \right) \otimes \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_c} \right), \tag{13}$$

where n_r and n_c are the number of rows and columns of the LR image and $\otimes$ denotes the standard Kronecker product.

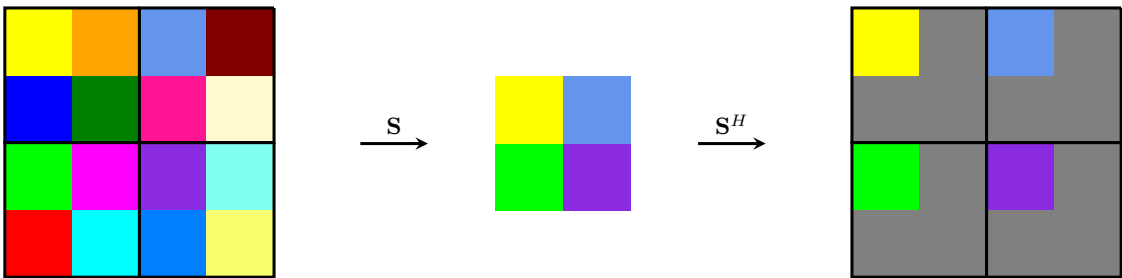Figure 3: **Action of the down-sampling and of the up-sampling operators S $\mathbf{S}^H$. S** acts on a 4×4 image with $L = 2$, $\mathbf{S}^H$ acts on a 2×2 image with $L = 2$.

On the structure of D. As far as the operator $\mathbf{D}$ is concerned, assuming periodic boundary conditions, we have that both matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ are BCCB, hence they can be diagonalized by Fourier transforms as

$$\mathbf{D}_x = \mathbf{F}^H \mathbf{\Sigma}_x \mathbf{F} \quad \text{and} \quad \mathbf{D}_y = \mathbf{F}^H \mathbf{\Sigma}_y \mathbf{F}, \tag{14}$$

with $\boldsymbol{\Sigma}_x \in \mathbb{R}^{N_h \times N_h}$ and $\boldsymbol{\Sigma}_y \in \mathbb{R}^{N_h \times N_h}$ diagonal matrices. As a consequence, the product $\mathbf{D}^H\mathbf{D} = \mathbf{D}_x^H\mathbf{D}_x + \mathbf{D}_y^H\mathbf{D}_y$, corresponding to the Laplace operator, can be expressed by

$$\mathbf{D}^H\mathbf{D} = \mathbf{F}^H\left(\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)\mathbf{F}. \tag{15}$$

Solving the ℓ_2-ℓ_2 substeps. We now focus our attention on the solution of the ℓ_2-ℓ_2 subproblems required for the computation of $\mathbf{u}^{k+1}$ both in (U-SISR) and (C-SISR). As mentioned above, this substep is the most expensive one in terms of computational times, as it requires, in the case of unstructured matrices, the use of iterative solvers, such as, e.g., Conjugate-Gradient schemes [2].

By optimality, the desired $\mathbf{u}^{k+1}$ is the solution of the following linear system

$$\left(\mathbf{H}^H\mathbf{S}^H\mathbf{S}\mathbf{H} + \beta^k\mathbf{D}^H\mathbf{D}\right)\mathbf{u}^{k+1} = \left(\mathbf{H}^H\mathbf{S}^H\mathbf{g} + \beta^k\mathbf{D}^H(\mathbf{z}^{k+1} - \frac{\boldsymbol{\lambda}^k}{\beta^k})\right). \tag{16}$$

To reduce the computational complexity of the linear system in (16) a direct solver is desirable. Due to the structure of the decimation operator $\mathbf{S}$, however, the product $\mathbf{S}\mathbf{H}$ cannot be diagonalized in the frequency domain, thus preventing any direct computation of $\mathbf{u}^{k+1}$ in terms of fast Fourier solvers. However, following [18] we can exploit the assumptions pointed out in Section 2 and manipulate (16) in terms of $\mathbf{F}$ and $\mathbf{F}^H$ to deduce the following chain of equalities

$$\left(\mathbf{F}^H\boldsymbol{\Lambda}^H\mathbf{F}\mathbf{S}^H\mathbf{S}\mathbf{F}^H\boldsymbol{\Lambda}\mathbf{F} + \beta^k\mathbf{F}^H\left(\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)\mathbf{F}\right)\mathbf{u}_{k+1} = \mathbf{r}^k \tag{17}$$

$$\left(\mathbf{F}^H\boldsymbol{\Lambda}^H\frac{1}{L^2}\left(\mathbf{J}_L \otimes \mathbf{I}_{n_r}\right) \otimes \left(\mathbf{J}_L \otimes \mathbf{I}_{n_c}\right)\boldsymbol{\Lambda}\mathbf{F} + \beta^k\mathbf{F}^H\left(\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)\mathbf{F}\right)\mathbf{u}_{k+1} = \mathbf{r}^k \tag{18}$$

$$\left(\boldsymbol{\Lambda}^H\frac{1}{L^2}\left(\mathbf{J}_L \otimes \mathbf{I}_{n_r}\right) \otimes \left(\mathbf{J}_L \otimes \mathbf{I}_{n_c}\right)\boldsymbol{\Lambda}\mathbf{F} + \beta^k\left(\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)\mathbf{F}\right)\mathbf{u}_{k+1} = \mathbf{F}\mathbf{r}^k \tag{19}$$

$$\left(\frac{1}{L^2}\underline{\boldsymbol{\Lambda}}^H\underline{\boldsymbol{\Lambda}} + \beta^k\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)\mathbf{F}\mathbf{u}_{k+1} = \mathbf{F}\mathbf{r}^k, \tag{20}$$

where we have defined

$$\mathbf{r}^k := \mathbf{H}^H\mathbf{S}^H\mathbf{g} + \beta^k\mathbf{D}^H\left(\mathbf{z}^{k+1} - \frac{\boldsymbol{\lambda}^k}{\beta^k}\right), \qquad \underline{\boldsymbol{\Lambda}} := \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_r}\right) \otimes \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_c}\right)\boldsymbol{\Lambda}. \tag{21}$$

From (20), we can thus deduce

$$\mathbf{u}^{k+1} = \mathbf{F}^H\left(\frac{1}{L^2}\underline{\boldsymbol{\Lambda}}^H\underline{\boldsymbol{\Lambda}} + \beta^k\boldsymbol{\Sigma_x}^H\boldsymbol{\Sigma_x} + \boldsymbol{\Sigma_y}^H\boldsymbol{\Sigma_y}\right)^{-1}\mathbf{F}\mathbf{r}^k. \tag{22}$$

By using now Woodbury formula [16] to determine the expression of $\mathbf{u}_{k+1}$ we have

$$\mathbf{u}^{k+1} = \frac{1}{\beta^k}\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{r}^k - \frac{1}{\beta^k}\mathbf{F}^H\boldsymbol{\Psi}\underline{\boldsymbol{\Lambda}}^H\left(\beta^k d\mathbf{I}_{N_l} + \underline{\boldsymbol{\Lambda}}\boldsymbol{\Psi}\underline{\boldsymbol{\Lambda}}^H\right)^{-1}\underline{\boldsymbol{\Lambda}}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{r}^k, \tag{23}$$

where $\boldsymbol{\Psi} := \mathbf{F}\left(\mathbf{D}^H\mathbf{D}\right)^{-1}\mathbf{F}^H$. In order to overcome the fact that the discrete Laplace operator $\mathbf{D}^H\mathbf{D}$ may not be invertible, we can follow [18] and add a regularization term $\sigma_{\boldsymbol{\mathcal{L}}}\|\mathbf{u}\|_2^2$, depending on a small constant $0 < \sigma_{\boldsymbol{\mathcal{L}}} \ll 1$, to make the operator $\boldsymbol{\Psi}$ invertible and the iteration (23) well-defined, so that $\boldsymbol{\Psi}_{\sigma_{\boldsymbol{\mathcal{L}}}} := \left(\boldsymbol{\Sigma}_x^H\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y^H\boldsymbol{\Sigma}_y + \sigma_{\boldsymbol{\mathcal{L}}}\mathbf{I}_{N_h}\right)^{-1}$.

Expression (23) provides now an efficient formula to compute at each $k \geq 1$ the quantity $\mathbf{u}^{k+1}$ since it only requires the inversion of diagonal matrices through standard FFT evaluations. We remark that such update is not possible for a general down-sampling operator $\mathbf{S}$ (such as, e.g., the Lanczos interpolation operator, often employed in the context of SISR problems), as the chain of equalities (10)-(13) is no longer true. Algorithm 3 reports the main step for the solution of the $\ell_2 - \ell_2$ subproblem (9).

Algorithm 3 – Fast solution of ℓ_2-ℓ_2 problems

input: $\mathbf{g} \in \mathbb{R}^{N_l}$, $\mathbf{S} \in \mathbb{R}^{N_l \times N_h}$, $\mathbf{H} \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{D}_x \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{D}_y \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{z}^{k+1} \in \mathbb{R}^{2N_h}$, $\beta^k > 0$, $0 < \sigma_{\boldsymbol{\mathcal{L}}} \ll 1$

output: $\mathbf{u}^{k+1}$

  1: $\mathbf{H} = \mathbf{F}^H \boldsymbol{\Lambda} \mathbf{F}$

  2: $\mathbf{D}_h = \mathbf{F}^H \boldsymbol{\Sigma}_x \mathbf{F}$

  3: $\mathbf{D}_v = \mathbf{F}^H \boldsymbol{\Sigma}_y \mathbf{F}$

     % *Compute* $\underline{\boldsymbol{\Lambda}}$ *and* $\boldsymbol{\Psi}$

  4: $\underline{\boldsymbol{\Lambda}} \leftarrow \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_r}\right) \otimes \left(\mathbf{1}_L^T \otimes \mathbf{I}_{n_c}\right) \boldsymbol{\Lambda}$

  5: $\boldsymbol{\Psi} \leftarrow \left(\boldsymbol{\Sigma}_x^H \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y^H \boldsymbol{\Sigma}_y + \sigma_{\boldsymbol{\mathcal{L}}} \mathbf{I}_{N_h}\right)^{-1}$

     % *Compute solution of the linear system*

  6: $\mathbf{r}^k \leftarrow \mathbf{H}^H \mathbf{S}^H \mathbf{g} + \beta^k \mathbf{D}^H(\mathbf{z}^{k+1} - \dfrac{\boldsymbol{\lambda}^k}{\beta^k})$

  7: $\mathbf{u}^{k+1} \leftarrow \dfrac{1}{\beta^k} \mathbf{F}^H \boldsymbol{\Psi} \mathbf{F} \mathbf{r}^k - \dfrac{1}{\beta^k} \mathbf{F}^H \boldsymbol{\Psi} \underline{\boldsymbol{\Lambda}}^H \left(\beta^k d\mathbf{I}_{N_l} + \underline{\boldsymbol{\Lambda}} \boldsymbol{\Psi} \underline{\boldsymbol{\Lambda}}^H\right)^{-1} \underline{\boldsymbol{\Lambda}}^H \boldsymbol{\Psi} \mathbf{F} \mathbf{r}^k$.

---

4 Numerical Results

In this section, we report several numerical experiments showing the performances of both considered algorithms for joint SISR and IP. All experiments reported in this section are conducted on a workstation Intel i5-6500, 8GB RAM using codes written in MATLAB R2019b and freely available as part of this work.

Test problems and evaluation of results. We simulate blurred and noisy LR acquisitions starting from ground-truth (GT) HR images by applying model (1). Unless otherwise specified, we set as default parameters the standard deviation of the Gaussian blur $\sigma_{\mathbf{H}} = 1$, the size of the Gaussian kernel equal to 9×9 and the standard deviation of the Gaussian noise $\sigma_{\boldsymbol{\eta}} = 0.01$. We denote by L the super-resolution factor. Following our previous notations, we thus have $N_h = L^2 \cdot N_l$.

The operators $\mathbf{H}$, $\mathbf{D}_x$ and $\mathbf{D}_y$ can be represented as discrete convolution operators, hence their action on HR images can be efficiently computed as an element-wise product between their Optical Transfer Function (OTF) and the 2D Fourier Transform of the HR image. To do so, we make use of the MATLAB functions `psf2otf` and `fft2` to compute the OTF of $\mathbf{H}$, $\mathbf{D}_x$ and $\mathbf{D}_y$ and the 2D Fourier Transform of the HR image, respectively.

Concerning the evaluation of the output image, we remark that we will not consider in the following common reconstruction-based Figures Of Merits (FOM), such as PSNR, SSIM since they are useful to compare the quality of the output image with the original one based on a comparison with the ground truth. Given the particular type of smoothing enforced by the ℓ_0 regularizer acting on the image gradient and the final IP objective our algorithms are expected not to provide outputs similar to the desired GT, but, rather, to provide a smoothed super-resolved image with simplified image contents.

Stopping criteria, regularization parameters and initializations. The unconstrained scheme iterations in (U-SISR) are stopped whenever the following relative stopping criterion

$$\frac{\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2}{\|\mathbf{u}^k\|_2} \le 0.0005,$$

is met. Concerning the stopping rule for the constrained scheme (C-SISR), we stop the algorithm when the constraint in (5) is satisfied with a tolerance of 5%

$$\|\mathbf{Du}^{k+1}\|_0 \le \alpha - 5\%\alpha.$$

We remark that the parameter $\alpha \in \mathbb{N}$ represents the number of desired jumps in the image gradient $\mathbf{Du}$ of the solution. Choosing α a priori is, typically, difficult. We found that a more practical parameter choice consists in choosing a certain fraction $\alpha_r = \frac{\alpha}{N} \in [0, 1]$ of the total possible number of jumps in the GT image (N is the number of pixels). Hence, in our experiments we fixed α_r as input parameter and derived α from it.

Regarding the choice of the initial guess $\mathbf{u}_0$ in both algorithms (U-SISR) and (C-SISR), we report in the following the convergence graphs corresponding to three different possible choices for $\mathbf{u}_0$: the image obtained from the LR image via Lanczos interpolation (case S1), the zero vector (case S2) and the solution of the ℓ_2-TV problem (case S3)

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}\in\mathbb{R}^{N_h}} \frac{1}{2}\|\mathbf{SHu} - \mathbf{g}\|_2^2 + \mu\|\mathbf{Du}\|_1. \tag{24}$$

As an example, in Figure 4 we show the HR image (a) and its LR ($L = 2$) counterpart (b), the results obtained by applying the constrained scheme (C-SISR) (Figures (c)-(d)) and the results of the unconstrained scheme (U-SISR) (Figures (e)-(f)) starting from S1 and S2.
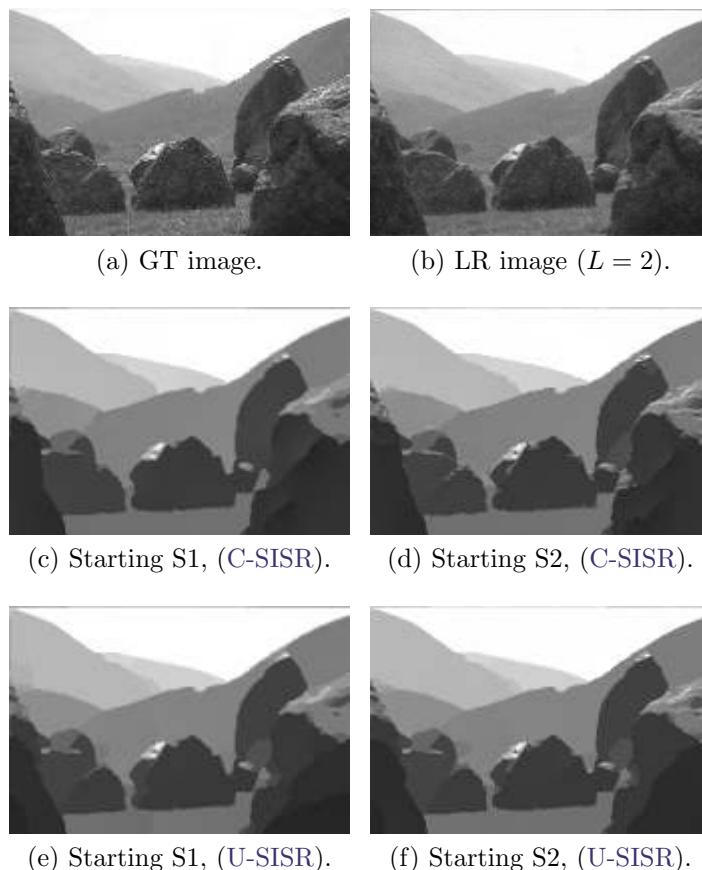

(a) GT image. (b) LR image ($L = 2$).

(c) Starting S1, (C-SISR). (d) Starting S2, (C-SISR).

(e) Starting S1, (U-SISR). (f) Starting S2, (U-SISR).

Figure 4: **HR images obtained from different starting guesses.** (a) GT image. (b) LR image (L=2). (d)-(e) Reconstructions with constrained scheme (C-SISR). (f)-(g) Reconstructions with unconstrained scheme (U-SISR).

To get more insights on the algorithm results depending on the starting guess, we report in Figure 5 some convergence plots obtained for different starting vectors $\mathbf{u}^0$. In each graphic, the red,

blue and yellow lines are associated to the starting choices S1, S2 and S3, respectively. We remark that due to the nonconvexity of the model, convergence to different local minimizers is expected. Convergence is shown here in terms both of the $\|\mathbf{Du}^k\|_0$ and of the objective function value (for the unconstrained model): we note that in terms of such quantities, solutions obtained with starting choices S1 and S3 empirically seem to converge to the same minimizers, while a different (suboptimal, as with higher final cost) solution is found corresponding to S2. Based on this preliminary tests, in the following, the algorithms are always initialized using the Lanczos interpolated image (case S1).
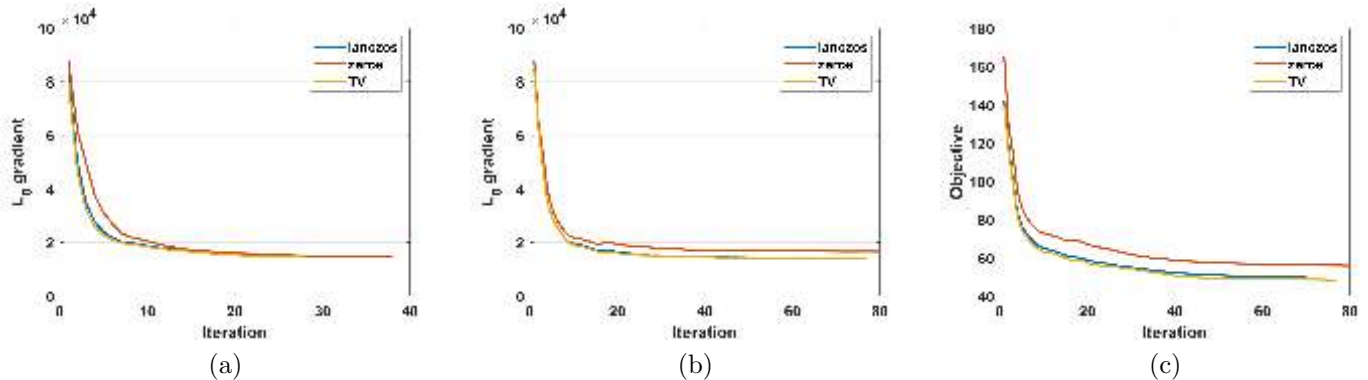

Figure 5: **Convergence plots for different initializations.** (a) $\|\mathbf{Du}^k\|_0$ vs. iteration number, (C-SISR). (b) $\|\mathbf{Du}^k\|_0$ vs. iteration number, (U-SISR). (c) Objective function vs. iteration number, (U-SISR). Red line: S1; blue line: S2; (c) yellow line: S3.

On the choice of the regularization parameter. To favor comparisons between the constrained and the unconstrained models, we fix the hyperparameter α_r related with the value α in the constrained model (7) and we estimate the regularization parameter μ of the unconstrained method so that the reconstruction computed by (U-SISR) has (approximately) the same number of jumps as the solution of the constrained model.

As an example, in Figure 6 we report two natural images of size 480×320 and their simulated LR acquisitions of size 240×160. We compute the HR solutions obtained by solving the constrained and the unconstrained models by setting $\alpha_r = \alpha/N = 0.16$ and choosing μ accordingly. We observe that the proposed choices of the hyperparamters α and μ produce comparable images.

4.1   Reconstruction Results

Super-resolution and segmentation of an object of interest. We apply both models to the photographic image of Figure 7(a), aiming as providing as a final result the SR and the segmentation of the bird in the foreground. We show the final HR images ($L = 2$) computed by the constrained/unconstrained algorithms with $\alpha_r = 0.083$ and $\mu = 0.022$, respectively. We notice that the results obtained are very similar and both methods provide a meaningful piecewise constant partitioning of the given image, as we can see also from the reported line profile.

Regularization effects. In the experiment reported in Figure 8 we increase the regularization strength in both algorithms. To do so, we decrease the value of the parameter α_r in the constrained scheme (C-SISR) from 0.25 to 0.07 (so as to promote less and less jumps in the final image) and compare with a corresponding choice for the unconstrained model (U-SISR) so as to have a similar value for $R_{\ell_0}(\mathbf{u}^*)$ in the final image. As previously noticed, the effect of increasing the proposed
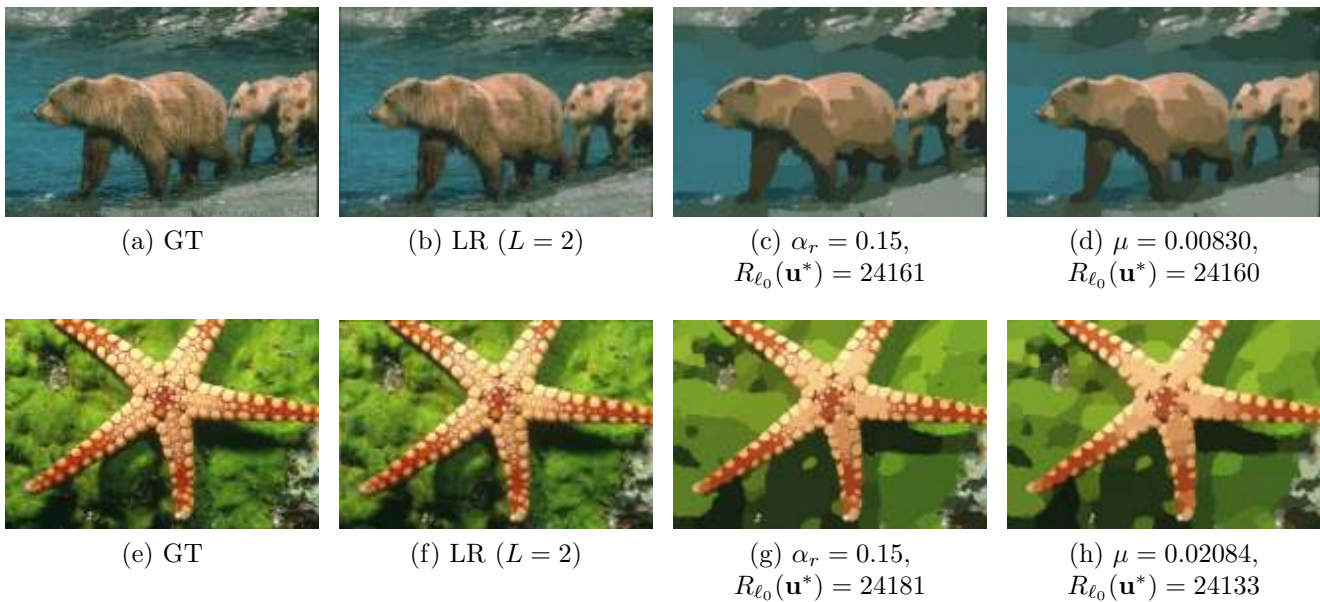
Dario Mylonopoulos, Pasquale Cascarano, Luca Calatroni, Elena Loli Piccolomini

Figure 6: **On the choice of μ and α.** Natural GT images of size 481×321 ((a) and (e)). Simulated LR acquisitions according to image formation model (1) by setting $L = 2$, $\sigma_\eta = 0.01$, $\sigma_H = 1.0$ ((b) and (f)). Reconstructions obtained by the constrained model (C-SISR) ((c) and (g)). Reconstructions obtained by the unconstrained model (U-SISR) ((d) and (h)).
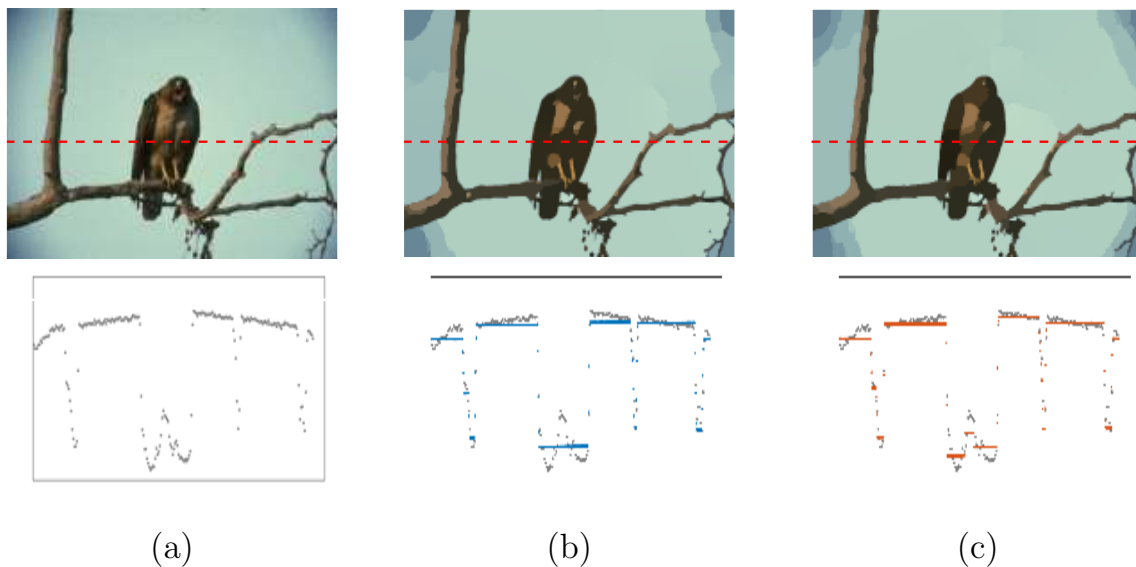

Figure 7: **Joint SR and segmentation of an object of interest**. (a) GT image. (b) Reconstruction obtained by using the constrained scheme (C-SISR). (c) Reconstruction obtained by using the unconstrained scheme (U-SISR). The plots correspond to the line profile of the red dashed line. Gray dots correspond to the GT image, while colored dots to the HR reconstructed images.

regularization is a reduced number of partitions in the restored image corresponding to a lower number of classes in a possible later classification.

Robustness to blur and noise degradation. In Figure 9 we analyse the effects of the blur degradation level, obtained by setting different values of the parameter σ_H, while considering no noise perturbation ($\sigma_\eta = 0$) and $L = 4$. In the first row of Figure 9, the LR images appear more and more
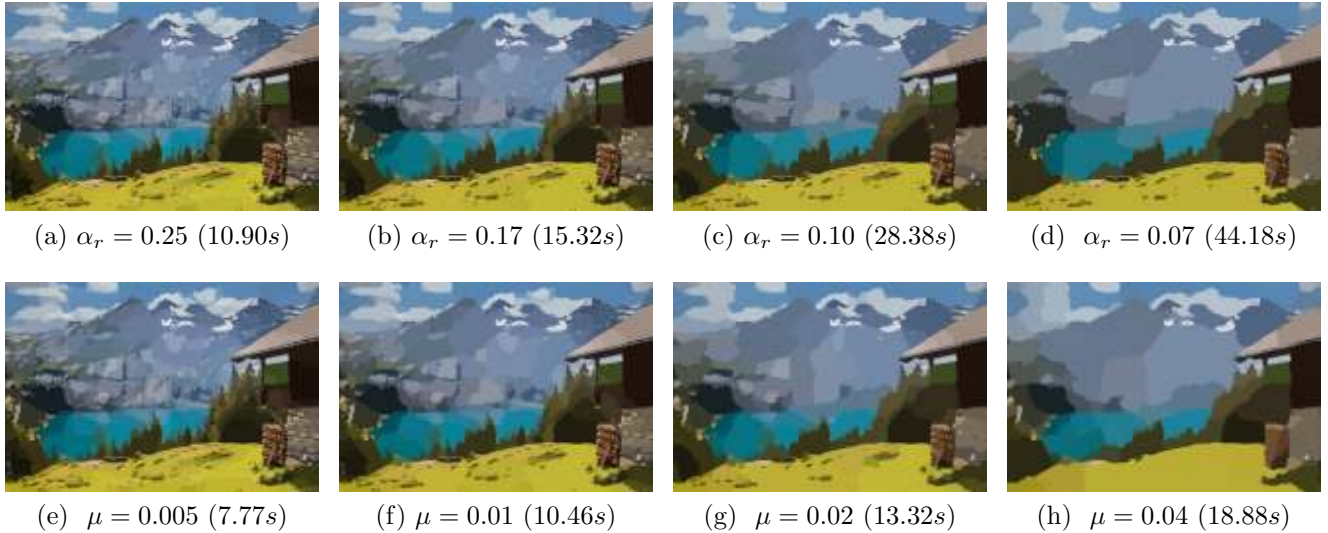
(a) $\alpha_r = 0.25$ (10.90s)   (b) $\alpha_r = 0.17$ (15.32s)   (c) $\alpha_r = 0.10$ (28.38s)   (d) $\alpha_r = 0.07$ (44.18s)

(e) $\mu = 0.005$ (7.77s)   (f) $\mu = 0.01$ (10.46s)   (g) $\mu = 0.02$ (13.32s)   (h) $\mu = 0.04$ (18.88s)

Figure 8: **Reconstructions with increasing regularization ($L = 2$).** (a)-(d) Images from constrained model (C-SISR). (e)-(h) Images from unconstrained model (U-SISR).

blurred from left to right. The reconstructions depicted in the second (constrained algorithm) and third (unconstrained algorithm) rows of the figure exhibit a very good visual quality but are slightly different, even if the number of jumps is almost the same: the constrained model appears sharper and preserving more details, while the unconstrained model partitions the image into larger regions and it is just more blurred. We argue that this trend may be linked to the different dependence on the thresholding parameters observed in Algorithm 1 and 2. While the gradient thresholding appearing in Algorithm 1 vanish for $\beta_k \to \infty$, the same does not happen for the analogous step in Algorithm 2, as it solely depends on the given parameter $\alpha > 0$. Late algorithmic iterations are thus expected not to be under-regularized for Algorithm 1, while still subject to regularization for Algorithm 2. Overall this may result in under-regularized solutions in the unconstrained case, where blur and noise are still slightly visible.

The masks reported in the last column confirm the effectiveness of the proposed super resolution method for image segmentation. In this case the C-SISR mask appears more accurate than the U-SISR.

Similarly, we also test the algorithms by considering the LR image corrupted only by noise of different intensities, without any blur ($\sigma_\mathbf{H} = 0$) and $L = 4$. In Figure 10 we report the HR results of the two proposed algorithms. Here all the results are very similar. This indicates that both approaches are robust with respect to the noise: the restoration is not degraded by the increasing noise affecting the LR input image. The masks shown in the last column on the C-SISR and U-SISR reconstructions appear very similar in this case; they both improve the mask obtained for the LR image.

Varying the super-resolution factor. Figure 11 contains the results obtained by varying factor L and with fixed blur/noise degradation levels corresponding to $\sigma_\mathbf{H} = 1.0, \sigma_\eta = 0.01$. In the first row of Figure 11 we show the LR images obtained from the GT (of size 480×320) with $L \in \{2, 3, 4\}$ and the results obtained by the constrained and unconstrained methods are displayed in the second and third row, respectively. The output of (C-SISR) and (U-SISR) are more and more different as L increases. As observed in the previous tests, the images computed by (C-SISR) preserve more details, whereas the reconstructions from (U-SISR) are less sharp and present larger flat regions. The masks computed on the reconstructions with $L = 4$ and reported in the last column of Figure 11
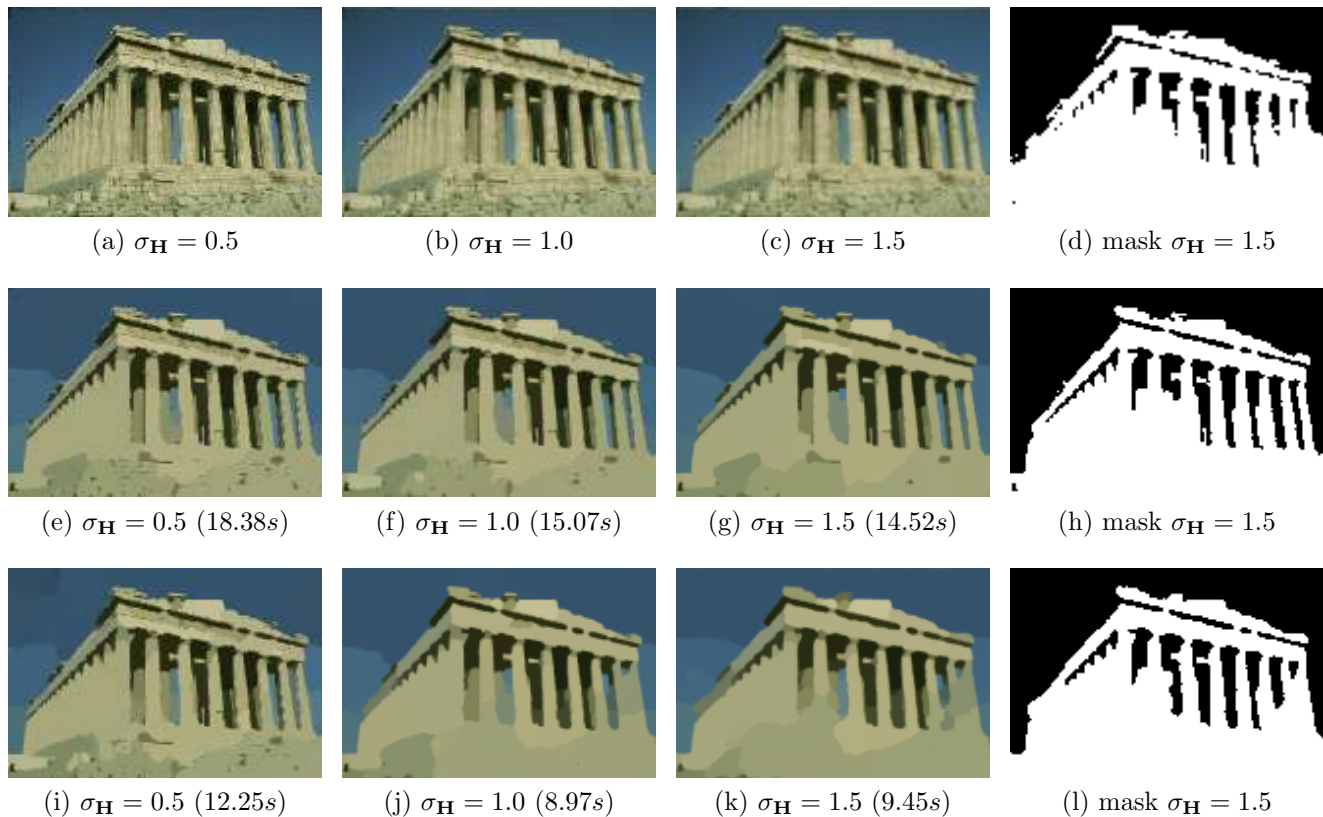
Dario Mylonopoulos, Pasquale Cascarano, Luca Calatroni, Elena Loli Piccolomini

Figure 9: **Reconstructions with increasing blur** ($L = 4$ **and** $\sigma_\eta = 0$**) and corresponding masks.** (a)-(d) LR image. (e)-(h) Constrained scheme (C-SISR) ($\alpha_r = 0.1$). (i)-(l) Unconstrained scheme (U-SISR) ($\mu = 0.03$).

show that the U-SISR method with the chosen regularization parameter over-smoothes the restored image, whereas the C-SISR method with the proposed number of jumps produces a very accurate segmentation mask.

Computational complexity. For most of the output images we have reported the computational time in seconds. We first observe that the executions are always very fast. In Figure 8 the time increases when $R_{\ell_0}(\mathbf{u}^*)$ decreases. In Figure 9 we notice that the algorithms are faster for increasing blur, whereas Figure 10 shows that the noise does not affect the algorithms run time.

As a final test, we report two graphs showing how the algorithms execution time scales with the size (i.e. the number of pixels of the GT image) and with factor L. Figure 12 displays the computational times (in seconds) for increasing number of pixels and for different values of L. We notice that the constrained method requires twice as much time as the unconstrained model to compute the desired result. The slopes of the displayed lines confirm a linear increasing of the time with respect to the image size in all the considered experiments.

5 Conclusions

We considered two joint single-image super-resolution (SISR) and image partitioning (IP) variational models based on the use of an ℓ_0-type jump penalizer combined with ℓ_2 data fidelity for favoring sharp gradient smoothing, which is desirable in reconstruction problems for both removing noise/blur degradation while preserving edges and for image simplification in view of subsequent analysis. Upon a specific assumption on the down-sampling operator (to be assumed as a decimation matrix) and by introducing suitable ADMM-based iterative schemes, we detail how the solution of the joint
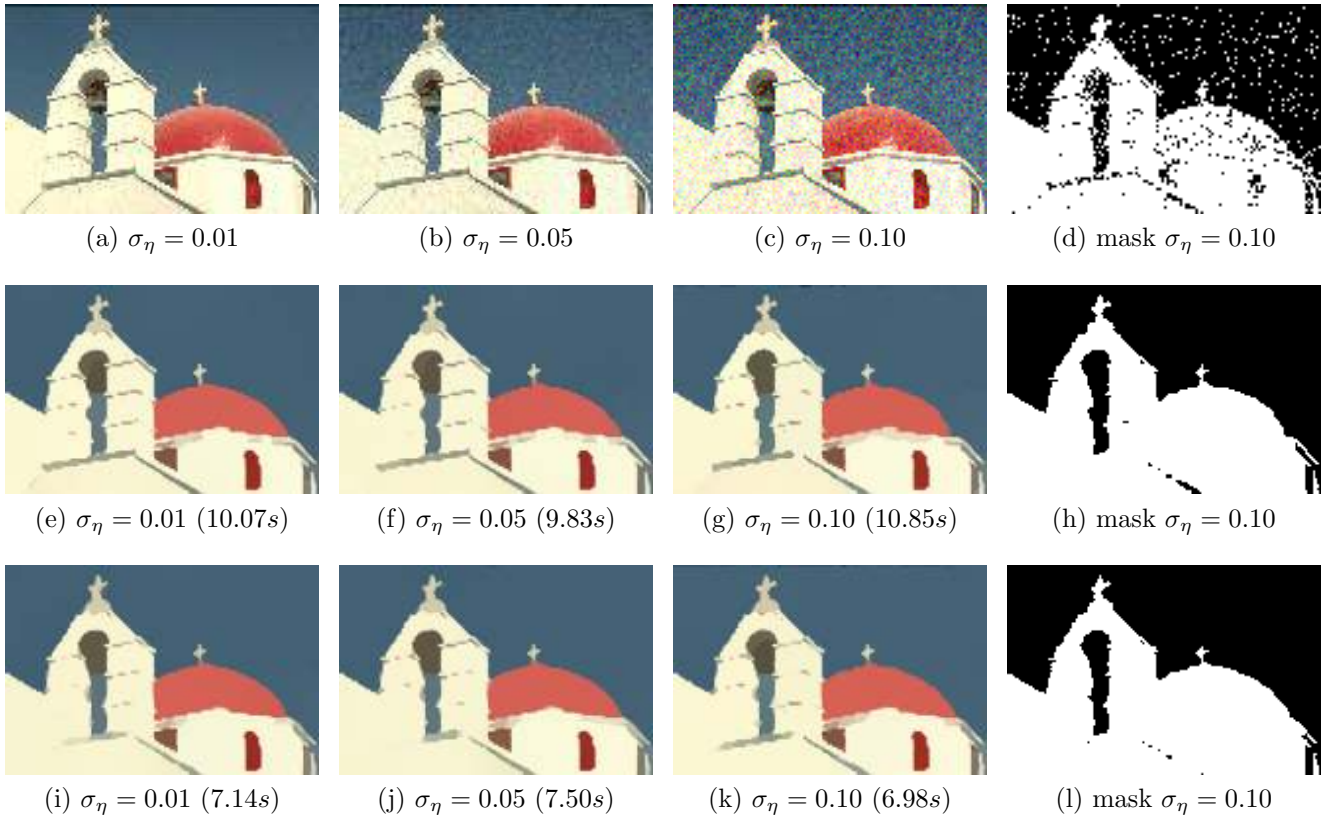
Figure 10: **Reconstructions with different level of noise ($L = 4$ and $\sigma_{\mathbf{H}} = 0$) and corresponding masks.** (a)-(d) LR image. (e)-(h) Constrained scheme (C-SISR) ($\alpha_r = 0.065$). (i)-(l) Unconstrained scheme (U-SISR) ($\mu = 0.03$).

approach can be computed very efficiently by diagonalization in the Fourier domain and via fast hard-thresholding/projections in the unconstrained/constrained case, respectively. Extensive numerical results confirming the robustness of both approaches to noise, blur degradation and upsampling size are reported. A MATLAB code implementing the constrained/unconstrained schemes is provided together with an online demo.
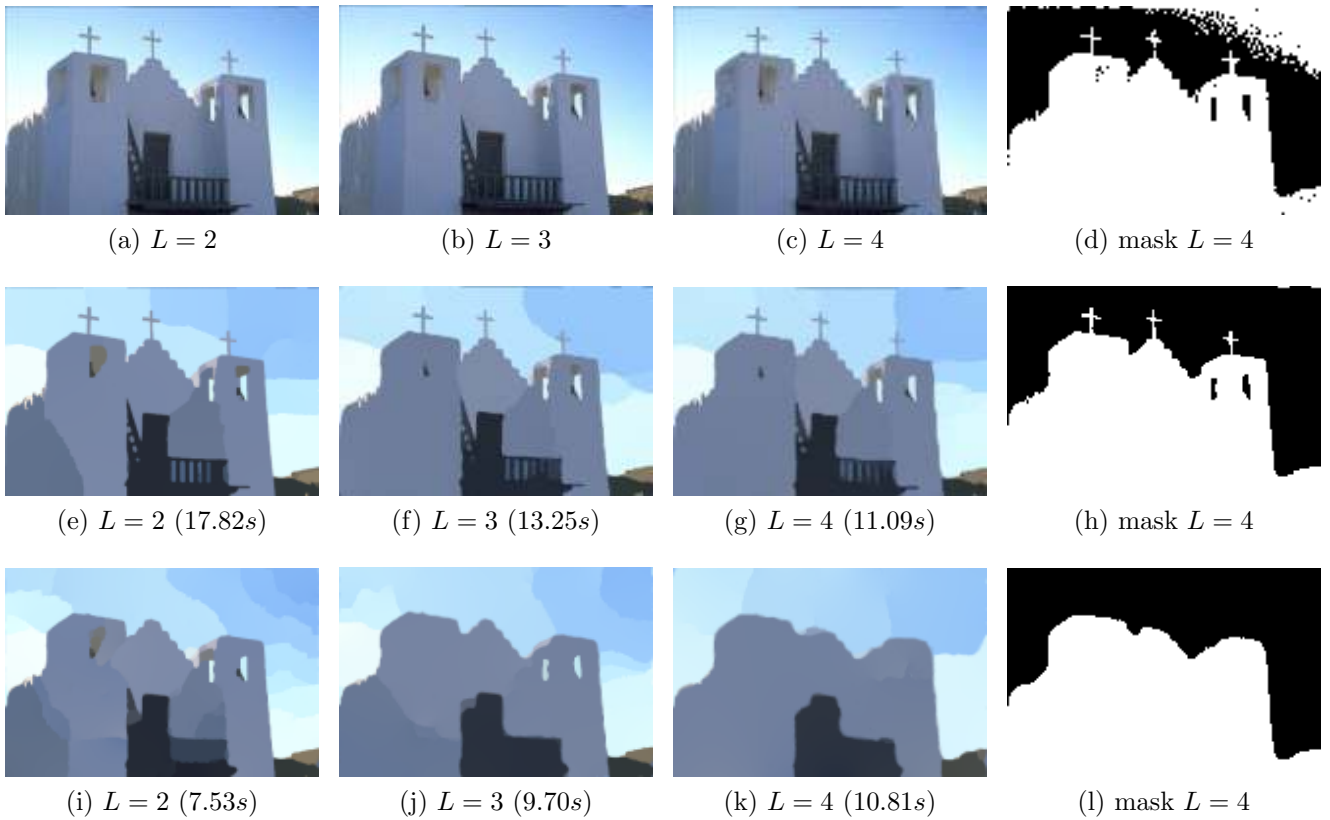
(a) $L = 2$ (b) $L = 3$ (c) $L = 4$ (d) mask $L = 4$

(e) $L = 2$ $(17.82s)$ (f) $L = 3$ $(13.25s)$ (g) $L = 4$ $(11.09s)$ (h) mask $L = 4$

(i) $L = 2$ $(7.53s)$ (j) $L = 3$ $(9.70s)$ (k) $L = 4$ $(10.81s)$ (l) mask $L = 4$

Figure 11: **Reconstructions with increasing super-resolution factor L ($\sigma_{\mathbf{H}} = 1.0, \sigma_\eta = 0.01$) and corresponding masks.** (a)-(d) LR image. (e)-(h) Constrained scheme (C-SISR) results ($\alpha_r = 0.065$). (i)-(l) Unconstrained scheme (U-SISR) results ($\mu = 0.03$).
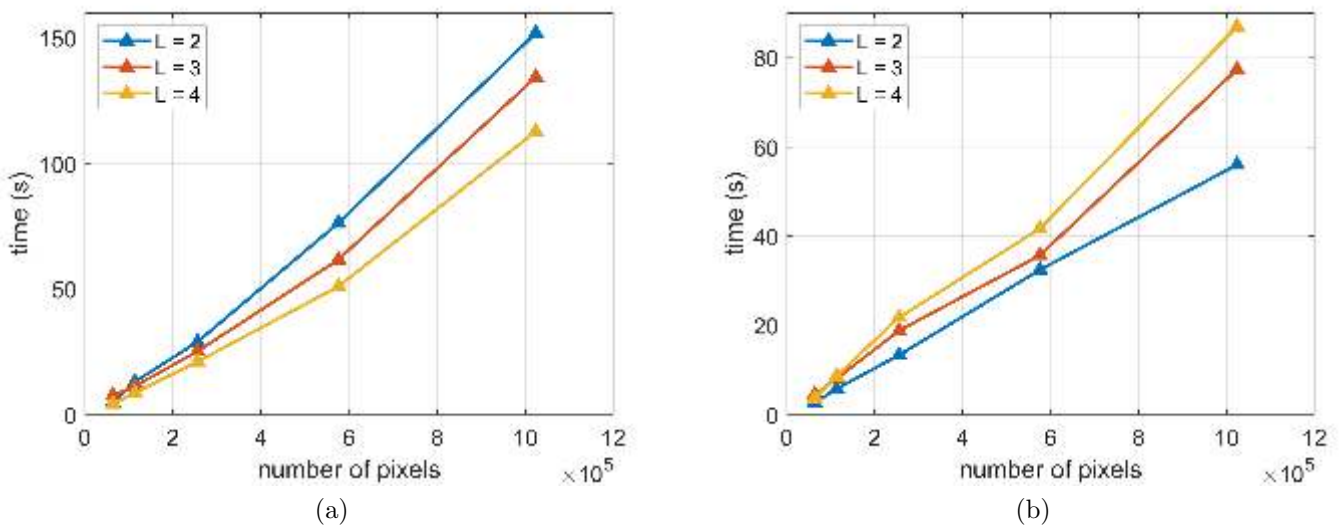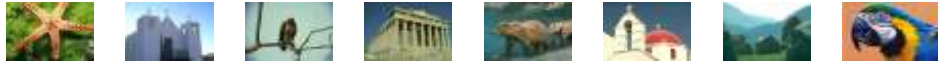

(a)   (b)

Figure 12: **Computational times for increasing image size.** Execution times (in seconds) vs. image size for the constrained (a) and unconstrained (b) models varying the super-resolution factor L.

6   Acknowledgments

Image Credits

        BSDS300 https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

 Pixabay https://pixabay.com/en/.

 EVICAN dataset https://edmond.mpdl.mpg.de/imeji/collection/l45s16atmi6Aa4sI

 provided by the authors.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Foundations and Trends in Machine Learning, 3 (2011), pp. 1–122. https://doi.org/10.1561/2200000016.

[2] P. Cascarano, L. Calatroni, and E. Loli Piccolomini, *Efficient ℓ^0 gradient-based super resolution for simplified image segmentation*, IEEE Transactions on Computational Imaging, (2021), pp. 1–1. https://doi.org/10.1109/TCI.2021.3070720.

[3] Q. Delannoy, C-H. Pham, C. Cazorla, C. Tor-Díez, G. Dollé, H. Meunier, N. Bednarek, R. Fablet, N. Passat, and F. Rousseau, *SegSRGAN: Super-resolution and segmentation using generative adversarial networksApplication to neonatal brain MRI*, Computers in Biology and Medicine, 120 (2020), p. 103755. https://doi.org/10.1016/j.compbiomed.2020.103755.

[4] L. Kiefer, M. Storath, and A. Weinmann, *PALMS: Image Partitioning-A New Parallel Algorithm for the Piecewise Affine-Linear Mumford-Shah Model*, Image Processing On Line, 10 (2020), pp. 124–149. https://doi.org/10.5201/ipol.2020.295.

[5] ——, *Iterative Potts minimization for the recovery of signals with discontinuities from indirect measurements: the multivariate case*, Foundations of Computational Mathematics, 21 (2021), pp. 649–694. https://doi.org/10.1007/s10208-020-09466-9.

[6] M. Mansouri and A. Mohammad-Djafari, *Joint super-resolution and segmentation from a set of low resolution images using a Bayesian approach with a Gauss-Markov-Potts prior*, International Journal of Signal and Imaging Systems Engineering, 3 (2010), pp. 211–221. https://dx.doi.org/10.1504/IJSISE.2010.038017.

[7] V.A. Morozov, *Methods for solving incorrectly posed problems*, Springer Science & Business Media, 2012. ISBN: 978-1-4612-5280-1.

[8] S. Ono, ℓ_0 *gradient projection*, IEEE Transactions on Image Processing, 26 (2017), pp. 1554–1564. http://dx.doi.org/10.1109/TIP.2017.2651392.

[9] M. Pragliola, L. Calatroni, A. Lanza, and F. Sgallari, *Residual whiteness principle for automatic parameter selection in ℓ_2-ℓ_2 image super-resolution problems*, in Scale Space and Variational Methods in Computer Vision, A. Elmoataz, J. Fadili, Y. Quéau, J. Rabin, and L. Simon, eds., Cham, 2021, Springer International Publishing, pp. 476–488.

[10] H. Shen, L. Zhang, B. Huang, and P. Li, *A MAP approach for joint motion estimation, segmentation, and super resolution*, IEEE Transactions on Image processing, 16 (2007), pp. 479–490. https://doi.org/10.1109/TIP.2006.888334.

[11] M. Storath, D. Rickert, M. Unser, and A. Weinmann, *Fast segmentation from blurred data in 3D fluorescence microscopy*, IEEE Transactions on Image Processing, 26 (2017), pp. 4856–4870. https://doi.org/10.1109/tip.2017.2716843.

[12] M. Storath, A. Weinmann, and L. Demaret, *Jump-sparse and sparse recovery using Potts functionals*, IEEE Transactions on Signal Processing, 62 (2014), pp. 3654–3666. https://doi.org/10.1109/TSP.2014.2329263.

[13] M. Storath, A. Weinmann, J. Frikel, and M. Unser, *Joint image reconstruction and segmentation using the Potts model*, Inverse Problems, 31 (2015), p. 025003. https://doi.org/10.1088/0266-5611/31/2/025003.

[14] N.K. Tuador, D.H. Pham, J. Michetti, A. Basarab, and D. Kouamé, *A novel fast 3d single image super-resolution algorithm*, in IEEE International Symposium on Biomedical Imaging (ISBI), 2021, pp. 73–76. https://doi.org/10.1109/ISBI48211.2021.9433809.

[15] Y. Wang, W. Yin, and J. Zeng, *Global Convergence of ADMM in Nonconvex Nonsmooth Optimization*, Journal of Scientific Computing, 78 (2019), pp. 29–63. https://doi.org/10.1007/s10915-018-0757-z.

[16] M.A. Woodbury, *Inverting modified matrices*, Statistical Research Group, 1950.

[17] L. Xu, C. Lu, Y. Xu, and J. Jia, *Image Smoothing via ℓ_0 Gradient Minimization*, ACM Transactions on Graphics (SIGGRAPH Asia), (2011). https://doi.org/10.1145/2024156.2024208.

[18] N. Zhao, Q. Wei, A. Basarab, N. Dobigeon, D. Kouamé, and J-Y. Tourneret, *Fast Single Image Super-Resolution Using a New Analytical Solution for l2-l2 Problems*, IEEE Transactions on Image Processing, 25 (2016), pp. 3683–3697. https://doi.org/10.1109/tip.2016.2567075.