Journal of Scientific Computing (2026) 109:71
https://doi.org/10.1007/s10915-026-03478-3

Low-Rank and Sparse Splitting for Glyph Extraction

Andrea Azzarelli[1] · Alessandro Buccini[1] · Anas El Hachimi[2] · Lothar Reichel[3]

Received: 13 January 2026 / Revised: 31 August 2026 / Accepted: 8 September 2026
© The Author(s) 2026

Abstract
This paper is concerned with digital extraction of archaeological glyphs from a surface. The input surface is decomposed into a background component that represents the underlying general shape and a sparse component that captures the fine details, including the glyphs. We separate these components by using a novel variational framework that promotes low-rank background matrices via the nuclear norm minimization. Depending on the parameters, the resulting optimization problem can be either convex or non-convex. In particular, by considering ℓ_q-quasi-norm sparsity penalties with $0 < q < 1$, we obtain non-convex models that promote sparsity and often enhance the separation of fine structures more than an ℓ_1-norm penalty. We evaluate the effectiveness of the proposed method when applied to synthetic as well as real-world data.

Keywords Glyph extraction · Variational models · Alternating direction method of multipliers (ADMM) · Low-rank approximation

Mathematics Subject Classification 65F22 · 65K10 · 65F99

1 Introduction

Consider a three-dimensional reconstruction of an archaeological surface obtained, e.g., through photometric stereo techniques; see, e.g., [11, 12] and references therein. We assume that the surface can be represented by a function $f\colon \mathbb{R}^2 \to \mathbb{R}$ such as a bas-relief. Let the matrix $D = [D_{i,j}] \in \mathbb{R}^{n_1 \times n_2}$ represent a discretization of the surface sampled on a

✉ Andrea Azzarelli
andrea.azzarelli@unica.it

Alessandro Buccini
alessandro.buccini@unica.it

Anas El Hachimi
anaselhachimi1997@gmail.com

Lothar Reichel
reichel@math.kent.edu

[1] Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy

[2] LMPA, Université du Littoral Cote d'Opale, 62228 Calais-Cedex, France

[3] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

Published online: 03 October 2026

uniform rectangular grid (x_i, y_j), $i = 1, 2, \ldots, n_1$, $j = 1, 2, \ldots, n_2$, so that each entry $D_{i,j} = f(x_i, y_j)$ corresponds to the surface height at the grid point (x_i, y_j).

Following the approach in [1, 14], we decompose D into two components

$$D = B + G,$$

where B represents the background surface and G is a sparse matrix that captures the glyphs. Thus, most entries of G vanish. A natural way to promote sparsity would be to minimize the ℓ_0-pseudo-norm of G, which counts the number of nonzero entries. However, the resulting optimization problem is combinatorial and generally NP-hard. Therefore, following standard convex relaxation techniques, the ℓ_0-pseudo-norm is replaced by the convex ℓ_1-norm, which promotes sparse solutions while leading to a tractable optimization problem. In [1] a splitting model that relies on the fractional Laplacian operator L^α was used. It is given by

$$\begin{aligned}(B^*, G^*) = \arg \min_{B,G\in\mathbb{R}^{n_1\times n_2}} \frac{1}{2} \left\| L^\alpha \left(\mathrm{vec}(B)\right)\right\|_2^2 + \mu \left\|G\right\|_1, \\ \text{such that } D = B + G,\end{aligned} \tag{1}$$

where $\mathrm{vec}(B)$ is the vectorization of the matrix B, L^α denotes the fractional Laplacian operator, $\|\cdot\|_2$ stands for the Euclidean vector norm, and

$$\|G\|_1 = \|\mathrm{vec}(G)\|_1 .$$

Thus, $\|G\|_1$ is the ℓ_1-norm of the vector whose entries are the elements of the matrix G. We will refer to this model as the Fractional Laplacian Splitting (`FLS`) model.

In the present work we propose an alternative variational framework for determining B and G based on low-rank matrix approximation. This choice is motivated by the observation that the underlying general shape can be accurately approximated by a low-rank matrix, whereas the fine details cannot. To support the low-rank modeling assumption, we consider a real archaeological surface without engraved glyphs. Figure 1 compares the original surface with its truncated SVD (TSVD) reconstruction obtained by retaining only the largest 10% of the singular values. The two surfaces are visually almost indistinguishable, indicating that the dominant geometric structure can be represented accurately by a low-rank approximation. This observation provides empirical evidence that the background component is well suited to a low-rank model.

A natural way to enforce low-rankness would be to minimize the rank function `rank`$(\cdot)$. However, rank minimization problems are generally NP-hard and computationally intractable. To circumvent this difficulty, we replace the rank function by its convex relaxation, namely the nuclear norm $\|\cdot\|_*$, defined as the sum of the singular values of the matrix; see, e.g., [9]. The nuclear norm also acts as a regularization term that promotes low-rank solutions while yielding a tractable optimization problem. In addition, we allow the ℓ_1-norm in the second term in the right-hand side of (1) to be an ℓ_q-quasi-vector-norm

$$\|z\|_q = \left(\sum_{i=1}^{n} |z_i|^q\right)^{1/q}, \qquad z = [z_1, z_2, \ldots, z_n]^T \in \mathbb{R}^n, \tag{2}$$

for some $0 < q < 1$. The function $\|\cdot\|_q$ is not a norm, because the triangle inequality is not satisfied. Nevertheless, it is useful for matrix extraction. The use of the ℓ_q-quasi-norm with $0 < q < 1$ is motivated by the fact that it promotes sparsity more strongly than the ℓ_1-norm and provides a better approximation of the ℓ_0-pseudo-norm. Thus, instead of solving the

problem (1), in the present work we will consider the minimization problem

$$(B^*, G^*) = \arg \min_{B,G\in\mathbb{R}^{n_1\times n_2}} \|B\|_* + \frac{\mu}{q}\|G\|_q^q, \quad 0 < q \leq 1. \\ \text{such that } D = B + G, \tag{3}$$

For simplicity, we henceforth will refer to the ℓ_q-quasi-norm as the ℓ_q-norm and define

$$\|G\|_q = \|\text{vec}(G)\|_q, \qquad 0 < q \leq 2. \tag{4}$$

We refer to the model (3) as the Low-Rank Splitting (LRS) model. It is known that minimizing with respect to the nuclear norm promotes solutions of low-rank. This suggests that the solution B^* of (3) is a low-rank approximation of the matrix D. In particular, B does not contain the fine details of the image that is represented by D; they will be represented by G, thereby isolating the glyphs. The model (3) addresses two of the main drawbacks of the model (1):

- The computations with L^α require that boundary conditions be imposed. To simplify the computations, periodic boundary conditions were imposed in [1]. However, these boundary conditions may result in artifacts close to the edges of the reconstructed image.
- There is no need to determine a parameter α. Its value affects the computed splitting in (1), but how a suitable value of α should be chosen is not obvious. The choice of the new parameter q is less critical, since the performance of the method is not very sensitive to the value of q.

A few related approaches have been proposed in the literature. In particular, a similar strategy was described in [8, 18]. Our proposed method differs from the one in [18] in the use of an ℓ_q-norm for some $0 < q < 1$, and from the method described in [8], because its authors are interested in the low-rank component only and assume that the sparse component has random uniformly distributed support. Moreover, our applications are different from those in [18], where the authors would like to split an image into a textured component and a cartoon component, while we are concerned with surfaces and do not assume any smoothness of the glyph. *GoDec* is another algorithm that aims to decompose a matrix into a low-rank component and a sparse component. It is described in [15, 23]. Also in these works the authors are only interested in the low-rank component and assume that bounds for the rank of the background matrix and for the number of non-zero elements in the sparse component are available before the the computations are begun.

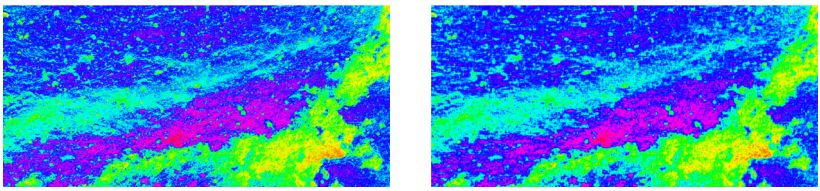
This paper is organized as follows. Section 2 discusses properties of the LRS model and the choice of the parameters μ and q in this model. The iterative method used for the solution of (3) is described in Sect. 3. Convergence of the iterations is shown in Sect. 4 and computed results are presented in Sect. 5. This section reports comparisons with already available methods and describes a real world application. Concluding remarks can be found in Sect. 6.

2 Motivation of the Model

This section discusses the model (3) as well as the choice of the parameters μ and q.

2.1 The Choice of ℓ_q-Norm

When seeking to compute a sparse solution of the minimization problem (3), one might consider using the norm $\|G\|_q^q$ for $q = 0$, because this norm counts the number of nonvanishing



(a) (b)

Fig. 1 A real archaeological surface without engraved glyphs $n_1 = 583$, $n_2 = 1241$. Panel **a** shows the original surface, panel **b** shows its TSVD reconstruction obtained by retaining the largest 58 singular values

entries of a vector. However, many algorithms that seek to determine a sparse solution replace the ℓ_0-norm of a vector with the ℓ_1-norm; see, e.g., [21]. A reason for this is that the ℓ_1-norm is convex and therefore easier to compute with. However, the ℓ_1-norm of a vector may be a poor approximation of the ℓ_0-norm of the same vector. To address this issue, we propose to use the ℓ_q-norm for some $0 < q < 1$ instead of the ℓ_1-norm, because the ℓ_q-norm furnishes a better approximation of the ℓ_0-norm; see [6, Figure 1] for an illustration. While minimization problems that use the ℓ_q-norm are non-convex, it is easier to compute an approximate solution for them than for minimization problems that use the ℓ_0-norm. Moreover, the ℓ_q-norm promotes sparsity better than the ℓ_1-norm by penalizing small coefficients more aggressively; see [5] for a discussion. The ℓ_q-norm is more sparsity promoting, the closer $1 > q > 0$ is to zero, however, the solution of the minimization problem (3) becomes more computationally demanding the smaller q is. Therefore, we choose a value of $q > 0$ that gives an approximate solution of (3) with an acceptable computational effort. We remark that the minimization problem (3) is non-smooth for $0 < q \leq 1$. Section 5 presents computed examples that compare the choices $q = 1$ and $q < 1$.

2.2 The Choice of Regularization Parameter

One of the main challenges when using a model of the form (3) is to choose an appropriate value of the regularization parameter μ. This parameter determines the relative importance of the fidelity and regularization terms. While there is no a priori known optimal choice, a reasonable initial choice is to determine μ so that the fidelity and regularization terms are of about the same magnitude. Hence, for problem (1), assuming that α is known, we may choose the regularization parameter

$$\overline{\mu} = \frac{\|L^{\alpha}\,\mathrm{vec}(D)\|_2^2}{\|D\|_1},$$

while for problem (3), assuming that q is known, we choose

$$\overline{\mu} = q\,\frac{\|D\|_*}{\|D\|_q^q}. \tag{5}$$

This formulation assumes that $\|L^{\alpha}\,\mathrm{vec}(D)\|_2^2$ and $\|D\|_*$ are fairly accurate approximations of the functionals evaluated on B. In view of the sparsity of G, this is a reasonable assumption. However, the assumption that $\|D\|_q^q$ is close to $\|G\|_q^q$ may not hold. In fact, we typically have $\|D\|_q^q > \|G\|_q^q$, which suggests that the parameter value (5) should be increased.

Empirically, we have found that good choices of the parameter μ often are in the interval $[2\,\overline{\mu},\, 10\,\overline{\mu}]$. While this does not resolve the issue of the choice of μ, it narrows the range of reasonable values.

3 An Iterative Solution Method

We determine matrices that approximate the solution $\{B^*, G^*\}$ of (3) by the Alternating Direction Method of Multipliers (ADMM). This method is described by Boyd et al. [3]. Since the q-norm is non-smooth for $0 < q \leq 1$, it is difficult to solve the minimization problem (3). We therefore use a smoothed version of this norm, similarly as in [17]. Thus, we replace (4) by

$$\|G\|_{q,\varepsilon}^q = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left(\sqrt{(G_{i,j})^2+\varepsilon^2}\right)^q, \qquad G = [G_{i,j}] \in \mathbb{R}^{n_1\times n_2},$$

for a small $\varepsilon > 0$. Therefore, the problem that we actually solve is

$$\begin{aligned}(B^*, G^*) = \arg \min_{B,G\in\mathbb{R}^{n_1\times n_2}} \|B\|_* + \frac{\mu}{q}\|G\|_{q,\varepsilon}^q\,,\\ \text{such that } D = B + G.\end{aligned} \tag{6}$$

The associated augmented Lagrangian is given by

$$\mathcal{L}_\rho(B, G, Y) = \|B\|_* + \frac{\mu}{q}\|G\|_{q,\varepsilon}^q + \langle Y, B + G - D\rangle + \frac{\rho}{2}\|B + G - D\|_2^2,$$

where $Y \in \mathbb{R}^{n_1\times n_2}$ is the Lagrange multiplier, $\langle\cdot,\cdot\rangle$ denotes the inner product

$$\langle Y, X\rangle = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} Y_{i,j}X_{i,j}, \qquad Y, X \in \mathbb{R}^{n_1\times n_2},$$

and $\rho > 0$ is a penalty parameter for constraint violation in the augmented Lagrangian formulation and also is a regularization parameter for the problem. We comment on the choice of ρ below in Sect. 4. The following algorithm describes an ADMM scheme for the solution of (6) by minimizing $\mathcal{L}_\rho$ alternately with respect to B and G, followed by updating the multiplier Y.

3.1 Algorithm 1: Solution of (6)

Initialization: Set $G = Y = O$, where $O \in \mathbb{R}^{n_1\times n_2}$ denotes the zero matrix.
Step 1: Update B
Minimize the augmented Lagrangian with respect to B while keeping G and Y fixed:

$$\begin{aligned}B^{(k+1)} &= \arg\min_B \mathcal{L}_\rho\left(B, G^{(k)}, Y^{(k)}\right)\\ &= \arg\min_B \|B\|_* + \frac{\rho}{2}\left\|B - \left(D - G^{(k)} - \frac{Y^{(k)}}{\rho}\right)\right\|_2^2.\end{aligned}$$

This is a standard nuclear norm minimization problem and can be solved by Singular Value Thresholding (SVT) [7]. Let

$$M^{(k)} = D - G^{(k)} - \frac{Y^{(k)}}{\rho}, \tag{7}$$

and compute the Singular Value Decomposition (SVD)

$$M^{(k)} = U^{(k)} \Sigma^{(k)} V^{(k)\,T}, \qquad \Sigma^{(k)} = \operatorname{diag}[\sigma_1, \ldots, \sigma_{\min\{n_1,n_2\}}].$$

Define the diagonal matrix $\tilde{\Sigma}^{(k)} \in \mathbb{R}^{n_1 \times n_2}$ with entries

$$\tilde{\sigma}_i = \max\left\{\sigma_i - \tfrac{1}{\rho}, 0\right\}, \qquad 1 \le i \le \min\{n_1, n_2\}$$

and let

$$B^{(k+1)} = U^{(k)} \tilde{\Sigma}^{(k)} V^{(k)\,T}.$$

To reduce the computational effort, we do not evaluate the complete SVD at each iteration as we only need the singular values larger than $1/\rho$. Instead, we compute a partial SVD. We detail this approach in Sect. 5.4.

Step 2: Update G

Solve the minimization problem

$$\begin{aligned} G^{(k+1)} &= \arg\min_G \mathcal{L}_\rho\left(B^{(k+1)}, G, Y^{(k)}\right) \\ &= \arg\min_G \frac{\mu}{q} \|G\|_{q,\varepsilon}^q + \frac{\rho}{2} \left\| G - \left(D - B^{(k+1)} - \frac{Y^{(k)}}{\rho}\right)\right\|_2^2. \end{aligned}$$

This corresponds to a proximal operator associated with the $\ell_{q,\varepsilon}$-quasi-norm,

$$G^{(k+1)} = \operatorname{prox}_{(\mu/\rho)\|\cdot\|_{q,\varepsilon}^q}\left(D - B^{(k+1)} - \frac{Y^{(k)}}{\rho}\right).$$

For $0 < q < 1$, this function is non-convex and there is no solution in closed form. However, to compute an approximate solution for the smoothed version. We use the adaptive method described in [6], but, we do not project into a Generalized Krylov Subspace. Therefore this method reduces to the one in [22]. In detail, we apply the following iterative scheme. Let

$$C^{(k)} = D - B^{(k+1)} - \frac{Y^{(k)}}{\rho},$$

and initialize with $\tilde{G}^{(0)} = C^{(k)}$.

For $h = 0, 1, \ldots$ until convergence, we iterate:

$$\tilde{W}^{(h)} = \left(\left(\tilde{G}^{(h)}\right)^2 + \varepsilon^2\right)^{\frac{q}{2}-1},$$

$$\tilde{G}^{(h+1)} = \frac{C^{(k)}}{1 + (\mu/\rho)\tilde{W}^{(h)}},$$

where all operations are elementwise. The iterations are terminated as soon as the difference between two successive iterates falls below a fixed tolerance $\tilde{\tau}$, or alternatively, after H iterations have been carried out. Then we set $G^{(k+1)} = \tilde{G}^{(h+1)}$. In the numerical experiments, we chose $\tilde{\tau} = 10^{-8}$ and $H = 100$.

Step 3: Update Y
The Lagrange multipliers are updated according to

$$Y^{(k+1)} = Y^{(k)} + \rho \left(B^{(k+1)} + G^{(k+1)} - D \right). \tag{8}$$

We stop iterating as soon as the matrices $B^{(k)}$ and $G^{(k)}$ do not change much with k, i.e., as soon as

$$\frac{\left\| B^{(k+1)} - B^{(k)} \right\|_2}{\left\| B^{(k)} \right\|_2} < \tau, \qquad \frac{\left\| G^{(k+1)} - G^{(k)} \right\|_2}{\left\| G^{(k)} \right\|_2} < \tau,$$

holds for a user-specified tolerance τ, or when the maximum number of iterations K is reached.

4 Convergence of the Iterations

We would like to show that the iterates $G^{(k)}$ determined by the algorithm described in Sect. 3 converge to a stationary point of the function (6). For simplicity we consider an equivalent formulation of the problem: We let $Z := D - B$ and solve

$$\begin{aligned} (Z^*, G^*) = \arg \min_{Z, G \in \mathbb{R}^{n_1 \times n_2}} & \|-Z + D\|_* + \frac{\mu}{q} \|G\|_{q,\varepsilon}^q \\ \text{such that } & Z = G. \end{aligned} \tag{9}$$

The corresponding augmented Lagrangian is

$$\mathcal{L}_\rho(Z, G; Y) = \|-Z + D\|_* + \frac{\mu}{q} \|G\|_{q,\varepsilon}^q + \langle Y, G - Z \rangle + \frac{\rho}{2} \|G - Z\|_2^2 .$$

We will show convergence of iterates generated by the ADMM by using an approach that is inspired by [4, 16] with several significant adjustments. The following assumption are required.

Assumption 1 Let $\left\{ G^{(k)} \right\}_{k \in \mathbb{N}}$ and $\left\{ Z^{(k)} \right\}_{k \in \mathbb{N}}$ be iterates generated by Algorithm 1. Assume that these sequences are bounded, i.e., there is a constant $R > 0$ such that $\left\| G^{(k)} \right\|_2 \leq R$ and $\left\| Z^{(k)} \right\|_2 \leq R$ for all $k \in \mathbb{N}$.

Assumption 2 The iterates $Z^{(k+1)}$ and $G^{(k+1)}$ satisfy

$$\begin{aligned} Z^{(k+1)} &= \arg\min_Z \mathcal{L}_\rho \left(Z, G^{(k)}; Y^{(k)} \right), \\ G^{(k+1)} &= \arg\min_G \mathcal{L}_\rho \left(Z^{(k+1)}, G; Y^{(k)} \right), \end{aligned}$$

i.e., the inner iterations converge to the unique minimizer of the respective subproblem.

Remark 1 Assumption 2 requires that ρ is large enough so that the subproblems are strictly convex and that the inner iterations reach convergence; see below. However, this requirement is rarely satisfied in practice since (i) only a finite number of inner iterations are carried out and these iterations may be terminated before convergence, and (ii) a large value of ρ may slow down the rate of convergence significantly. To ensure that the algorithm is computationally feasible, a small value of ρ may have to be used. The subproblems solved therefore might not be convex. Even when a stationary point is computed, this may be a saddle point or a local

minimum. Nevertheless, vast numerical experience suggests that the algorithm converges when accurate enough approximate minimizations are carried out and when ρ is not too large. In all numerical experiments presented in Sect. 5, we set $\rho = 1$.

Define

$$g(Z) = \|Z - D\|_*, \qquad h_q(G) = \frac{\mu}{q} \|G\|_{q,\varepsilon}^q .$$

Lemma 1 *Let $0 < q < 2$. Then the gradient ∇h_q is Lipschitz continuous, i.e.,*

$$\left\| \nabla h_q(X_1) - \nabla h_q(X_2) \right\| \le L \|X_1 - X_2\|, \quad X_1, X_2 \in \mathbb{R}^{n_1 \times n_2}$$

for some constant $L > 0$ that is independent of X_1 and X_2.

Proof For simplicity, let the dimension be $n = 1$. Then

$$h_q'(x) = \mu x (x^2 + \varepsilon^2)^{\frac{q}{2}-1}.$$

To show that the function $h_q'(x)$ is Lipschitz continuous, we demonstrate that its derivative is bounded. Extension to higher dimensions is straightforward.

We have

$$\begin{aligned} h''(x) &= \mu \left[2 \left(\frac{q}{2} - 1 \right) x^2 (x^2 + \varepsilon^2)^{\frac{q}{2}-2} + (x^2 + \varepsilon^2)^{\frac{q}{2}-1} \right] \\ &= \mu (x^2 + \varepsilon^2)^{\frac{q}{2}-2} \left[(q-2)x^2 + (x^2 + \varepsilon^2) \right] \\ &= \mu (x^2 + \varepsilon^2)^{\frac{q}{2}-2} \left[(q-1)x^2 + \varepsilon^2 \right] \sim \mu (q-1) x^{q-2} \underset{x \to \infty}{\longrightarrow} 0. \end{aligned}$$

If $0 < q < 2$, then the continuous function $h''(x)$ approaches a constant at infinity and, therefore, is bounded. □

Before showing our main result, we need some auxiliary results and definitions.

Lemma 2 *Consider the iterates $Y^{(k)}$ and $G^{(k)}$, and let Assumption 2 hold. Then*

$$\left\| Y^{(k+1)} - Y^{(k)} \right\|_2 \le L \left\| G^{(k+1)} - G^{(k)} \right\|_2, \quad \forall\, k \in \mathbb{N},$$

where L is the Lipschitz constant in Lemma 1.

Proof By Assumption 2, $G^{(k+1)}$ is a minimizer of $\mathcal{L}_\rho \left(Z^{(k+1)}, G; Y^{(k)} \right)$. Therefore, we have

$$\begin{aligned} 0 &= \nabla \left(h_q(G) + \langle Y^{(k)}, G \rangle + \frac{\rho}{2} \left\| G - Z^{(k+1)} \right\|_2^2 \right) \Big|_{G^{(k+1)}} \\ &= \nabla h_q \left(G^{(k+1)} \right) + Y^{(k)} + \rho \left(G^{(k+1)} - Z^{(k+1)} \right). \end{aligned}$$

From this equality and the definition of $Y^{(k+1)}$ in (8), it follows that

$$\nabla h_q \left(G^{(k+1)} \right) = -Y^{(k+1)}. \tag{10}$$

This relation and the Lipschitz continuity of ∇h_q imply that

$$\left\| Y^{(k+1)} - Y^{(k)} \right\|_2 = \left\| -\nabla h_q(G^{(k+1)}) + \nabla h_q(G^{(k)}) \right\|_2 \le L \left\| G^{(k+1)} - G^{(k)} \right\|_2,$$

which concludes the proof. □

We will require the notion of subgradient.

Definition 1 (*Subgradient*) Let $D \subseteq \mathbb{R}^n$ and let $g\colon D \to \mathbb{R}$ be a convex continuous function. The *subgradient* of g in X_0 is defined as

$$\partial g(X_0) = \left\{ V \in \mathbb{R}^n \mid g(X) - g(X_0) \geq \langle V, X - X_0 \rangle \quad \forall X \in \mathbb{R}^{n_1 \times n_2} \right\}.$$

For discussions on subgradients and a proof of the following proposition, we refer to [10, 13, 19, 20].

Proposition 1 *Let $g(X)$ be a convex continuous function and let $h(X)$ be a differentiable function, both defined in an open set $D \subset \mathbb{R}^{n_1 \times n_2}$. The subgradient of the function $g + h$ at $X_0 \in D$ is the set of vectors obtained as a sum of an element of the subgradient of g and the gradient of h,*

$$\partial(g + h)(X_0) = \partial g(X_0) + \nabla h(X_0).$$

We recall the definition of an ω-strongly convex function.

Definition 2 A continuous function f is said to be strongly convex with parameter $\omega > 0$ if the following inequality holds for all points X_1, X_2 in its domain and for all $V \in \partial f(X)$:

$$f(X_1) - f(X_2) \leq \langle V, X_1 - X_2 \rangle - \frac{\omega}{2} \|X_1 - X_2\|^2.$$

We are in a position to show the following result.

Proposition 2 *Under Assumption 2 and with the notation above, assume that the parameter $\rho > 0$ is large enough so that*

$$G \to h_q(G) + \frac{\rho}{2} \|G - Z\|_2^2 \quad \text{and} \quad Z \to \|-Z + D\|_* + \frac{\rho}{2} \|G - Z\|_2^2$$

are ω-strongly convex functions of G and Z, respectively. Then

$$\begin{aligned} &\mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k+1)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\ &\leq \left(\frac{L^2}{\rho} - \frac{\omega}{2}\right) \left\| G^{(k+1)} - G^{(k)} \right\|_2^2 - \frac{\omega}{2} \left\| Z^{(k+1)} - Z^{(k)} \right\|_2^2. \end{aligned}$$

Proof We add and subtract the same quantity to obtain

$$\begin{aligned} &\mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k+1)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\ &= \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k+1)}\right) - \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k)}\right) \\ &\quad + \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right). \end{aligned} \tag{11}$$

Some manipulations of the first two terms in the right-hand side of (11) and using the definition (8) of $Y^{(k+1)}$ yields

$$\begin{aligned} &\mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k+1)}\right) - \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k)}\right) \\ &= \left\langle Y^{(k+1)}, Z^{(k+1)} - G^{(k+1)} \right\rangle - \left\langle Y^{(k)}, Z^{(k+1)} - G^{(k+1)} \right\rangle \\ &= \left\langle Y^{(k+1)} - Y^{(k)}, Z^{(k+1)} - G^{(k+1)} \right\rangle \end{aligned}$$

$$= \left\langle Y^{(k+1)} - Y^{(k)}, \frac{1}{\rho}(Y^{(k+1)} - Y^{(k)}) \right\rangle = \frac{1}{\rho} \left\| Y^{(k+1)} - Y^{(k)} \right\|_2^2. \tag{12}$$

Consider the last two terms in the right-hand side of (11). Applying the ω-strong convexity of $\mathcal{L}_\rho$ gives

$$\begin{aligned}
&\mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\
&= \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k)}\right) - \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k)}; Y^{(k)}\right) \\
&\quad + \mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k)}; Y^{(k)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\
&\overset{(a)}{\leq} \left\langle \nabla \mathcal{L}_\rho\left(G^{(k+1)}\right), G^{(k+1)} - G^{(k)} \right\rangle - \frac{\omega}{2} \left\| G^{(k+1)} - G^{(k)} \right\|_2^2 \\
&\quad + \left\langle \partial \mathcal{L}_\rho\left(Z^{(k+1)}\right), Z^{(k+1)} - Z^{(k)} \right\rangle - \frac{\omega}{2} \left\| Z^{(k+1)} - Z^{(k)} \right\|_2^2 \\
&\overset{(b)}{\ni} -\frac{\omega}{2}\left(\left\| G^{(k+1)} - G^{(k)} \right\|_2^2 + \left\| Z^{(k+1)} - Z^{(k)} \right\|_2^2 \right),
\end{aligned} \tag{13}$$

where inequality (a) follows from Proposition 1 and holds for each element in the subgradient. Therefore, inclusion (b) follows from $\nabla \mathcal{L}_\rho\left(G^{(k+1)}\right) = 0$ and $0 \in \partial \mathcal{L}_\rho\left(Z^{(k+1)}\right)$, which follow from Assumption 2 and Proposition 1. Then summing (12) and (13) yields

$$\begin{aligned}
&\mathcal{L}_\rho\left(Z^{(k+1)}, G^{(k+1)}; Y^{(k+1)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\
&\leq \frac{1}{\rho} \left\| Y^{(k+1)} - Y^{(k)} \right\|_2^2 - \frac{\omega}{2}\left(\left\| G^{(k+1)} - G^{(k)} \right\|_2^2 + \left\| Z^{(k+1)} - Z^{(k)} \right\|_2^2 \right) \\
&\leq \frac{L^2}{\rho} \left\| G^{(k+1)} - G^{(k)} \right\|_2^2 - \frac{\omega}{2}\left(\left\| G^{(k+1)} - G^{(k)} \right\|_2^2 + \left\| Z^{(k+1)} - Z^{(k)} \right\|_2^2 \right),
\end{aligned}$$

where the second inequality follows from Lemma 2. Rearranging this expression concludes the proof. □

We can now show that the sequence $\left\{\mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right)\right\}_{k\in\mathbb{N}}$ converges to a limit point as k increases.

Lemma 3 *Under the assumptions and notation of Proposition 2, assume that ρ is large enough so that $\frac{L^2}{\rho} - \frac{\omega}{2} < 0$ and such that*

$$f(X) = h_q(X) + \left(\frac{\rho}{2} - L^2\right) \left\| X - Z^{(k)} \right\|_2^2 \tag{14}$$

is ω-strongly convex for all k. Note that such a choice of ρ is possible because $\omega(\rho)$ is a monotonically increasing function of ρ. Define

$$v = \inf_Z \left\{ g(Z) + h_q(Z) \right\}.$$

Then

$$\lim_{k\to\infty} \mathcal{L}_\rho\left(G^{(k)}, Z^{(k)}; Y^{(k)}\right) \geq v,$$

and the sequence $\left\{\mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right)\right\}_{k\in\mathbb{N}}$ converges.

Proof If $\frac{L^2}{\rho} - \frac{\omega}{2} < 0$, then the expressions $\mathcal{L}_\rho(Z^{(k)}, G^{(k)}; Y^{(k)})$ are monotonically decreasing as k increases. We now show that these expressions are bounded from below. It follows from the proof of Lemma 2 that $Y^{(k)} = -\nabla h_q\left(G^{(k)}\right)$. Substituting this expression into the definition of $\mathcal{L}_\rho$, we obtain that the augmented Lagrangian evaluated at $\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right)$ satisfies

$$
\begin{aligned}
&\mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\
&= g(Z^{(k)}) + h_q(G^{(k)}) + \left\langle -\nabla h_q(G^{(k)}), G^{(k)} - Z^{(k)} \right\rangle + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&= g(Z^{(k)}) + h_q(G^{(k)}) - \left\langle \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \\
&\quad + \left\langle \nabla h_q(Z^{(k)}) - \nabla h_q(G^{(k)}), G^{(k)} - Z^{(k)} \right\rangle + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&= g(Z^{(k)}) + h_q(G^{(k)}) - \left\langle \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \\
&\quad - \left\langle \nabla h_q(G^{(k)}) - \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&\geq g(Z^{(k)}) + h_q(G^{(k)}) - \left\langle \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \\
&\quad - \left| \left\langle \nabla h_q(G^{(k)}) - \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \right| + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&\geq g(Z^{(k)}) + h_q(G^{(k)}) - \left\langle \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \\
&\quad - \left\| \nabla h_q(G^{(k)}) - \nabla h_q(Z^{(k)}) \right\|_2 \left\| G^{(k)} - Z^{(k)} \right\|_2 + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&\geq g(Z^{(k)}) + h_q(G^{(k)}) - \left\langle \nabla h_q(Z^{(k)}), G^{(k)} - Z^{(k)} \right\rangle \\
&\quad - L^2\left\| G^{(k)} - Z^{(k)} \right\|_2^2 + \frac{\rho}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&= g(Z^{(k)}) + h_q(G^{(k)}) + \left\langle \nabla h_q(Z^{(k)}), Z^{(k)} - G^{(k)} \right\rangle + \left(\frac{\rho}{2} - L^2\right)\left\| G^{(k)} - Z^{(k)} \right\|_2^2 \\
&= g(Z^{(k)}) + \left\langle \nabla h_q(Z^{(k)}), Z^{(k)} - G^{(k)} \right\rangle + f(G^{(k)}),
\end{aligned}
$$

where in the last equality we introduced the ω-strongly convex function f defined in (14). We observe that

$$
\begin{aligned}
f(Z^{(k)}) &= h_q(Z^{(k)}), \\
\nabla f(Z^{(k)}) &= \nabla h_q(Z^{(k)}) - (\rho - 2L^2)(Z^{(k)} - Z^{(k)}) = \nabla h_q(Z^{(k)}).
\end{aligned}
$$

The ω-strong convexity of f implies

$$
f(G^{(k)}) + \left\langle \nabla f(Z^{(k)}), Z^{(k)} - G^{(k)} \right\rangle \geq f(Z^{(k)}) + \frac{\omega}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2
$$

and, therefore, using the definition of f, we get

$$
\begin{aligned}
&h_q(G^{(k)}) + \left(\frac{\rho}{2} - L^2\right)\left\| G^{(k)} - Z^{(k)} \right\|_2^2 + \left\langle \nabla h_q(Z^{(k)}), Z^{(k)} - G^{(k)} \right\rangle \\
&\geq h_q(Z^{(k)}) + \frac{\omega}{2}\left\| G^{(k)} - Z^{(k)} \right\|_2.
\end{aligned}
$$

Combining the above inequalities yields

$$\begin{aligned}\mathcal{L}_\rho\left(G^{(k)}, Z^{(k)}; Y^{(k)}\right) &\geq g(Z^{(k)}) + h_q(Z^{(k)}) + \frac{\omega}{2}\left\|G^{(k)} - Z^{(k)}\right\|_2^2 \\ &\geq g(Z^{(k)}) + h_q(Z^{(k)}) \geq \nu.\end{aligned}$$

Since the sequence $\left\{\mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right)\right\}_{k\in\mathbb{N}}$ is monotonically decreasing and bounded from below, it is convergent. □

We now show some properties of the sequences $\left\{G^{(k)}\right\}_{k\in\mathbb{N}}$ and $\left\{Z^{(k)}\right\}_{k\in\mathbb{N}}$.

Lemma 4 *With the notation and assumptions of Lemma 3, the sequences* $\left\{G^{(k)}\right\}_{k\in\mathbb{N}}$ *and* $\left\{Z^{(k)}\right\}_{k\in\mathbb{N}}$ *satisfy*

$$\sum_{j=0}^{\infty}\left\|G^{(j+1)} - G^{(j)}\right\|_2^2 < \infty \quad \text{and} \quad \sum_{j=0}^{\infty}\left\|Z^{(j+1)} - Z^{(j)}\right\|_2^2 < \infty.$$

Moreover, they admit converging subsequences.

Proof Let k be arbitrary but fixed. It follows from Proposition 2 that

$$\begin{aligned}&\mathcal{L}_\rho\left(Z^{(0)}, G^{(0)}; Y^{(0)}\right) - \mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right) \\ &\geq -\sum_{j=0}^{k-1}\left[\left(\frac{L^2}{\rho} - \frac{\omega}{2}\right)\left\|G^{(j+1)} - G^{(j)}\right\|_2^2 - \frac{\omega}{2}\left\|Z^{(j+1)} - Z^{(j)}\right\|_2^2\right] \geq 0,\end{aligned}$$

where the last inequality is a consequence of the assumption that $\frac{L^2}{\rho} - \frac{\omega}{2} < 0$. Let $\mathcal{L}_\rho^*$ denote the limit point of the sequence $\left\{\mathcal{L}_\rho\left(Z^{(k)}, G^{(k)}; Y^{(k)}\right)\right\}_{k\in\mathbb{N}}$. The existence of a limit follows from Lemma 3. Then

$$\begin{aligned}&\sum_{j=0}^{\infty}\left[\left(\frac{\omega}{2} - \frac{L^2}{\rho}\right)\left\|G^{(j+1)} - G^{(j)}\right\|_2^2 + \frac{\omega}{2}\left\|Z^{(j+1)} - Z^{(j)}\right\|_2^2\right] \\ &\leq \mathcal{L}_\rho\left(Z^{(0)}, G^{(0)}; Y^{(0)}\right) - \mathcal{L}_\rho^*.\end{aligned}$$

Therefore,

$$\sum_{j=0}^{\infty}\left\|G^{(j+1)} - G^{(j)}\right\|_2^2 < \infty \quad \text{and} \quad \sum_{j=0}^{\infty}\left\|Z^{(j+1)} - Z^{(j)}\right\|_2^2 < \infty.$$

The existence of convergent subsequences follows from Assumption 1. □

We can show our main result.

Theorem 1 *Under the assumptions above and with the same notation, let* (Z^*, G^*) *denote the limit point of a converging subsequence of* $(Z^{(k)}, G^{(k)})$*. Then it holds*

(i) $G^* = Z^*$*;*
(ii) Let $G^{(j_k)}$ *denote a converging subsequence. Then the sequence* $\left\{Y^{(j_k)}\right\}_{j_k\in\mathbb{N}}$ *converges to a limit point* Y^* *such that* $\nabla h_q(G^*) + Y^* = 0$ *and* $0 \in \partial g(Z^*) - Y^*$*;*
(iii) G^* *is a stationary point of* $g + h_q$*, i.e.,* $0 \in \partial g(G^*) + \nabla h_q(G^*)$.

Proof First observe that the existence of the limit point (Z^*, G^*) is ensured by Lemma 4. We can now show properties (i)-(iii) of the theorem.

(i) It follows from Lemma 4 that $\left\| G^{(j_k)} - G^{(j_k-1)} \right\|_2 \to 0$ as $k \to \infty$. Therefore, Lemma 2 yields $\left\| Y^{(j_k)} - Y^{(j_k-1)} \right\|_2 \to 0$ as $k \to \infty$. It follows from the definition (8) of $Y^{(j_k)}$ that

$$\lim_{k\to\infty} \left\| G^{(j_k)} - Z^{(j_k)} \right\|_2 = \lim_{k\to\infty} \left\| Y^{(j_k)} - Y^{(j_k-1)} \right\|_2 = 0.$$

This implies $G^* = Z^*$.

(ii) Recall that by (10), we have $Y^{(j_k)} = -\nabla h_q \left(G^{(j_k)} \right)$. Therefore,

$$Y^* = \lim_{j_k\to\infty} -\nabla h_q \left(G^{(j_k)} \right) = -\nabla h_q \left(G^* \right),$$

where the last equality follows from the continuity of ∇h_q. This shows that the sequence $\left\{ Y^{(j_k)} \right\}_{j_k \in \mathbb{N}}$ converges and the first relation in the statement follows.

The optimality condition of $Z^{(j_k)}$ implies that

$$\begin{aligned} 0 &\in \partial g \left(Z^{(j_k)} \right) - Y^{(j_k-1)} - \rho \left(G^{(j_k-1)} - Z^{(j_k)} \right) \\ &= \partial g \left(Z^{(j_k)} \right) - Y^{(j_k)} + \rho \left(G^{(j_k)} - G^{(j_k-1)} \right). \end{aligned}$$

We obtain the second relation in the statement by taking the limit $k \to \infty$.

(iii) Since $G^* = Z^*$, adding the two equalities in (ii), shows that

$$\nabla h_q(G^*) + \partial g(G^*) \ni 0.$$

This concludes the proof. □

5 Numerical Results

This section describes results obtined by Algorithm 1 as well as by other methods. We first discuss the construction of the test data. The test surfaces are made up of three additive components:

- *A discretized smooth surface S*. This component consists of large areas with varying curvature, including convex and concave zones that feature relative minima, maxima, and saddle points. In our experiments, we define the function S on a uniformly spaced grid (x_i, y_j) in $[-1, 1] \times [-1, 1]$, where $i = 1, 2, \ldots, n_1$ and $j = 1, 2, \ldots, n_2$.
- *A sparse glyph* G^*. This component represents a simulated incision carved into the surface S.
- *A rough component R*. Since engravings are typically made on non-polished materials, we introduce a rough component to simulate surface irregularities. This component is constructed by generating a random matrix E with entries drawn independently from a standard normal distribution. This matrix is then smoothed by convolution with either a Gaussian filter (with standard deviation σ) or, alternatively, by a motion blur filter to simulate the presence of directional striations, as are commonly observed on eroded rock surfaces. To restrict the roughness to regions outside the glyph, we enforce $E_{i,j} = 0$ wherever $G^*_{i,j} \neq 0$. The final rough component is formed as

$$R = \nu \|G^* + S\|_2 E,$$

Table 1 Errors and computing time for Experiment 1

		Gaussian		motion	
		FLS	LRS	FLS	LRS
$2\overline{\mu}$	$f_1(G)$	0.87	1.49	1.91	1.73
	$f_2(G)$	0.65	0.68	1.27	0.90
	cpu time (s)	**5.83**	**2.07**	**0.56**	**2.94**
$5\overline{\mu}$	$f_1(G)$	0.83	0.55	2.00	1.37
	$f_2(G)$	0.70	0.47	1.41	0.68
	cpu time (s)	**3.68**	**5.64**	**0.49**	**5.83**
$10\overline{\mu}$	$f_1(G)$	1.10	1.15	2.00	1.55
	$f_2(G)$	0.96	0.93	1.41	1.19
	cpu time (s)	**2.25**	**4.82**	**0.47**	**14.63**

CPU times are highlighted in bold

where $\nu \in [0, 1]$ is a parameter that controls the roughness intensity, and $\|\cdot\|_2$ denotes the Frobenius norm.

We obtain D by

$$D = S + G^* + R.$$

Following [1], we evaluate the quality of the reconstructed glyph by comparing its normalized version to the ground truth. For $p \in \{1, 2\}$, we define the normalized error metric as

$$f_p(G) = \left\| \frac{G}{\|G\|_p} - \frac{G^*}{\|G^*\|_p} \right\|_p. \tag{15}$$

This metric measures the discrepancy in shape between the two matrices G and G^*, independently of their scaling.

5.1 Experiment 1: Comparison between FLS and LRS

In this section we compare the approaches (1) and (3) with $q = 1$. The exact glyph for the first example is a simple triangular shape carved into a background surface

$$S_{i,j} = (0.3)e^{-\frac{x_i^2}{2}} - (0.2)\sin(\pi y_j) - 0.2$$

of dimension $n_1 = n_2 = 560$. Two different kinds of rough background surfaces are considered: *Gaussian* and *motion*. Figure 2 shows the two starting surface seen from above colored by height.

We solved the two problems with both the FLS and LRS methods. The stopping parameters used for the algorithms were set to $K = 200$ and $\tau = 10^{-3}$. Figure 3 shows the computed reconstructions and Table 1 allows us to compare the relative errors of the two methods and their computing time. We can observe that in the "Gaussian" example, FLS was able to reconstruct the glyph with satisfactory accuracy, but the best result is obtained with LRS and $\mu = 5\overline{\mu}$. The LRS algorithm is in most of the examples slower because of the computation of the SVD at each step.

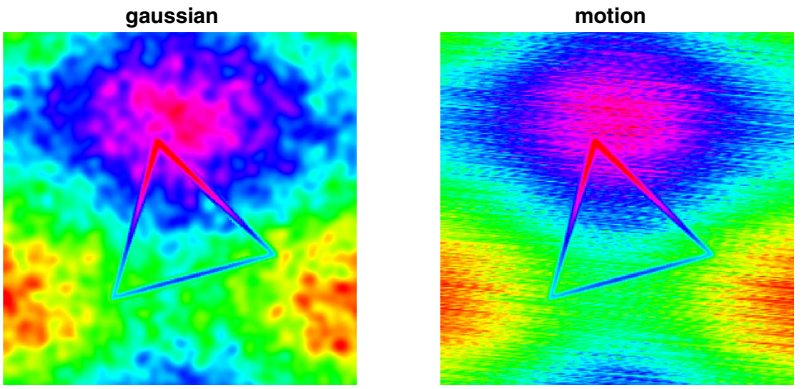


Fig. 2 Initial surfaces D. Panel **a** shows the example for which the roughness component is generated by a Gaussian filter with standard deviation $\sigma = 8$; panel **b** displays the example for which the roughness component is generated by using a motion blur filter with an angle of $-5°$. In both cases, the roughness intensity parameter is set to $\nu = 0.2$

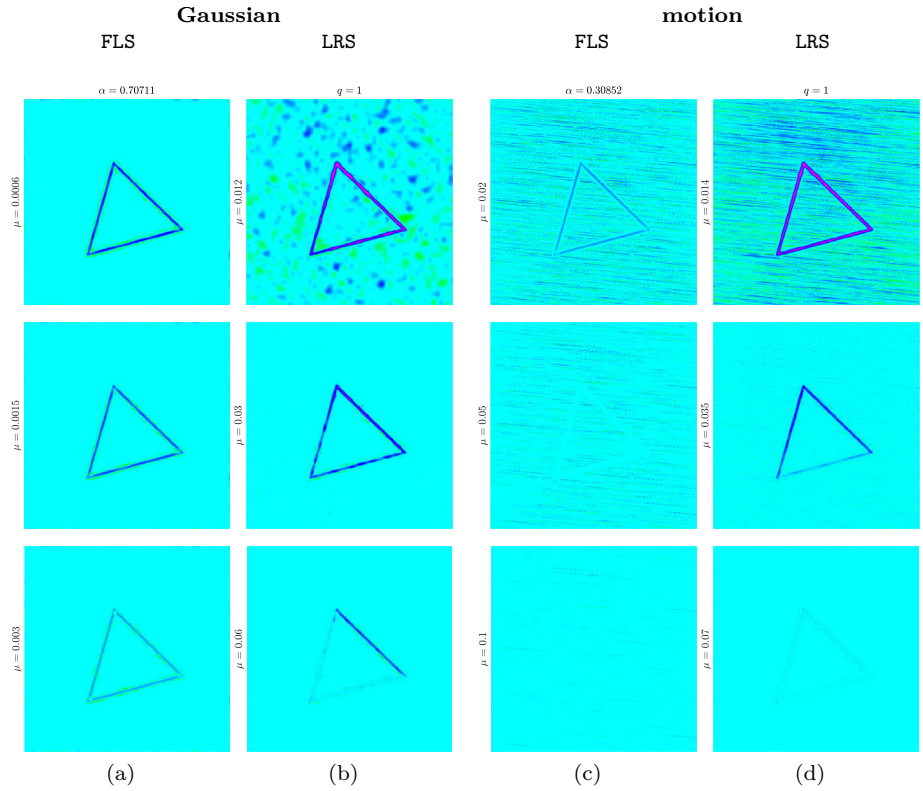
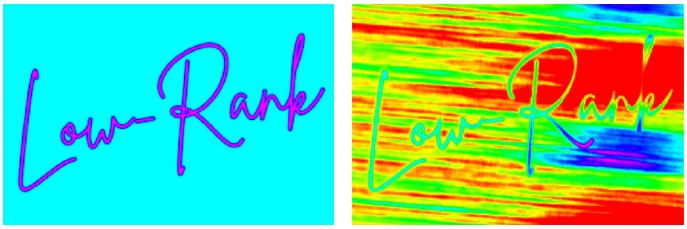


Fig. 3 Reconstructed glyphs G for Experiment 1. Comparison between `FLS` and `LRS` with $q = 1$



(a) (b)

Fig. 4 Panel **a** shows the exact glyph for Experiment 2. Panel **b** displays the starting image D. It is generated by using motion blur with angle $-5°$. The roughness intensity parameter is set to $\nu = 0.25$

Table 2 Errors and computing time for Experiment 2

		$q = 0.3$	$q = 0.5$	$q = 0.8$	$q = 1$
$2\overline{\mu}$	$f_1(G)$	1.14	0.58	1.07	1.31
	$f_2(G)$	0.72	0.47	0.60	0.72
	cpu time (s)	**9.99**	**10.06**	**10.26**	**0.07**
$3\overline{\mu}$	$f_1(G)$	0.55	0.62	0.68	0.95
	$f_2(G)$	0.46	0.57	0.54	0.63
	cpu time (s)	**10.04**	**9.77**	**10.18**	**0.07**
$4\overline{\mu}$	$f_1(G)$	0.44	0.98	0.82	0.84
	$f_2(G)$	0.43	0.77	0.69	0.68
	cpu time (s)	**10.92**	**9.86**	**9.79**	**0.06**
$5\overline{\mu}$	$f_1(G)$	0.62	1.44	1.09	0.95
	$f_2(G)$	0.58	0.99	0.85	0.78
	cpu time (s)	**9.87**	**10.84**	**14.24**	**0.06**

CPU times are highlighted in bold

5.2 Experiment 2: Testing Different Values of *q*

The model (3) allows the choice of a parameter $0 < q < 1$. This choice is helpful when it is difficult to isolate the glyph by only changing the value of μ. The effectiveness of the solution for the case $q = 1$ has already been demonstrated in [8, 18]. Here, we illustrate that letting $q < 1$ may give improved results. The exact glyph and the starting image for the second example are shown in Fig. 4. In this example $n_1 = 200$, $n_2 = 300$, and the background surface is defined by

$$S_{i,j} = (0.3)e^{-0.4(3x_i-2)^2} \sin\left(\tfrac{3}{2}\pi y_i\right) - 0.2, \quad \begin{matrix} i = 1, 2, \ldots, n_1, \\ j = 1, 2, \ldots, n_2. \end{matrix}$$

We compared results obtained with four different choices of the parameters $q \in \{0.3, 0.5, 0.8, 1\}$, $\mu \in \{2\overline{\mu}, 3\overline{\mu}, 4\overline{\mu}, 5\overline{\mu}\}$ with $\overline{\mu}$ computed as in (5) for each value of q.

This experiment illustrates that allowing $q < 1$ leads to better results; in fact the best reconstruction is obtained for $q = 0.3$ and $\mu = 4\overline{\mu}$ according to both measures of the error f_1 and f_2; see Table 2. This is in agreement with visual inspection; see Fig. 5. The main drawback is that the computation is more time consuming than for $q = 1$ due to the iterations required for the computation of $G^{(k+1)}$.

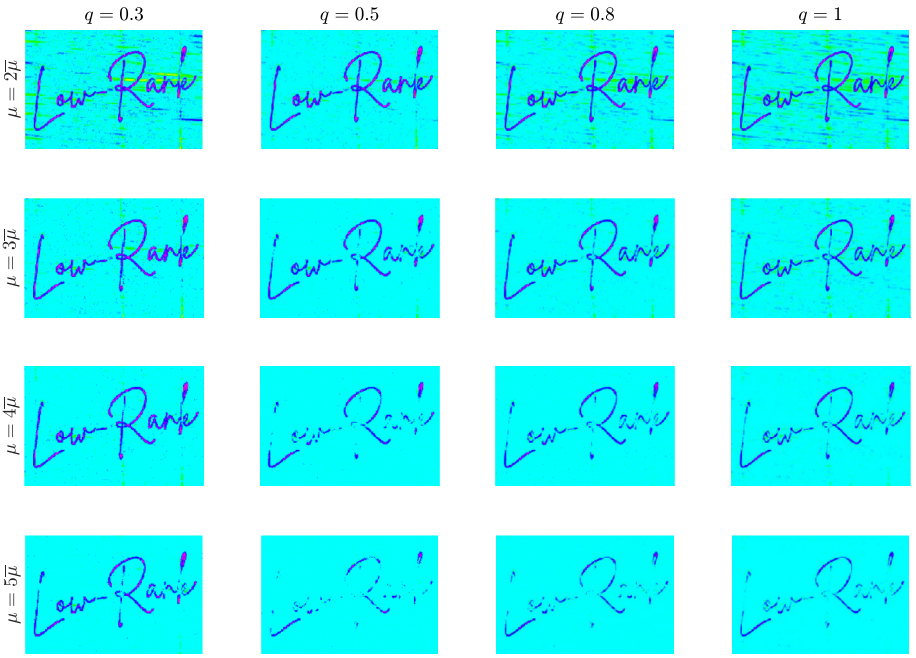


Fig. 5 Reconstructed glyphs G for Experiment 2. Testing LRS for different values of q

5.3 Experiment 3: Comparison with the GoDec Algorithm

In this experiment we compare our algorithm with the GoDec algorithm described in [23]. The latter algorithm requires a user to specify upper bounds for the rank of B and for the number of non-zero elements of G before the beginning of the computations, respectively denoted r and N_z. We tested the two algorithms on matrices of size $n = n_1 = n_2 = 100$, and varied the rank r of the underlying surface matrix S as well as the roughness level ν. Three roughness levels $\nu \in \{0.02, 0.1, 0.4\}$ and six ranks $r \in \{1, 2, 3, 5, 8, 10\}$ were considered.

To generate surfaces of different ranks, we computed $S_r = UC_rV^T$, where the columns of $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ are discretized sinusoidal functions of increasing frequency. The matrix $C_r = \text{diag}(c_1, \ldots, c_r, 0, \ldots, 0)$, with $c_i = 10^{-16\frac{i-1}{n-1}}$ for $i = 1, \ldots, n$, has entries logarithmically spaced between 1 and 10^{-16}. The exact glyph G^* is taken from software package [1]. Figure 6 shows an example of the data matrix D.

The parameters for the LRS method are set to $q = 0.1$ and $\mu = 5\overline{\mu}$, while for GoDec the exact rank of the underlying surface r and the number of non-zero entries in the true sparse component G^*, increased by 30%, are provided as upper bounds.

As shown in Table 3, both methods are able to achieve good reconstructions in almost all cases. Although GoDec has access to more information about the exact solution, LRS obtains better results in the majority of cases. Indeed, as can be seen more clearly in Fig. 7, the ratio $f_p(G_{\text{LRS}})/f_p(G_{\text{GoDec}})$ is smaller than 1 in almost all examples for both $p = 1$ and $p = 2$.

In terms of computing time, LRS is faster for small and medium values of ν, and comparable to GoDec for $\nu = 0.4$.

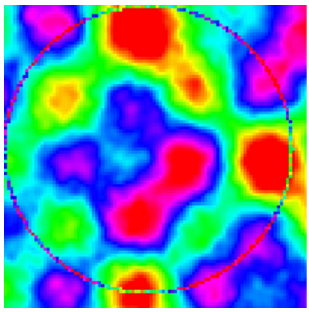


Fig. 6 Example of test surface for Experiment 3 with $r = 5$, $\nu = 0.1$

Table 3 Results of Experiment 3

		$\nu = 0.02$		$\nu = 0.10$		$\nu = 0.40$	
		LRS	GoDec	LRS	GoDec	LRS	GoDec
$r = 1$	$f_1(G)$	0.02	0.06	0.09	0.04	0.32	0.18
	$f_2(G)$	0.03	0.09	0.07	0.04	0.18	0.17
	cpu time (s)	**0.05**	**0.54**	**0.04**	**0.13**	**0.05**	**0.02**
$r = 2$	$f_1(G)$	0.04	0.19	0.09	0.17	0.73	0.39
	$f_2(G)$	0.05	0.25	0.08	0.23	0.43	0.39
	cpu time (s)	**0.05**	**0.35**	**0.05**	**0.08**	**0.08**	**0.08**
$r = 3$	$f_1(G)$	0.02	0.49	0.08	0.14	0.59	0.77
	$f_2(G)$	0.04	0.64	0.09	0.13	0.47	0.69
	cpu time (s)	**0.07**	**0.61**	**0.08**	**0.24**	**0.11**	**0.08**
$r = 5$	$f_1(G)$	0.13	0.28	0.11	0.25	0.63	1.02
	$f_2(G)$	0.20	0.35	0.16	0.33	0.60	0.87
	cpu time (s)	**0.13**	**0.63**	**0.09**	**0.19**	**0.11**	**0.07**
$r = 8$	$f_1(G)$	0.23	0.43	0.27	0.39	0.51	1.23
	$f_2(G)$	0.37	0.55	0.40	0.47	0.61	1.00
	cpu time (s)	**0.08**	**0.71**	**0.07**	**0.09**	**0.06**	**0.10**
$r = 10$	$f_1(G)$	0.22	0.43	0.27	0.50	0.76	1.09
	$f_2(G)$	0.35	0.53	0.40	0.59	0.74	0.92
	cpu time (s)	**0.06**	**0.79**	**0.11**	**0.08**	**0.12**	**0.08**

Comparison of LRS and GoDec. The smallest scores in terms of $f_p(G)$ and the smallest times are underlined CPU times are highlighted in bold

5.4 Accelerating the SVD Computations

Figure 8 shows that the number of singular values required for the computation of $B^{(k)}$ depends strongly on the parameter μ and tends to stabilize for k large enough, usually around $k \approx 10$. We took advantage of this behavior to reduce the computational effort of the algorithm by only determining a few of the largest singular values as follows. This number is determined by information from the previous five iterations. During the first five iterations, the full SVD is computed.

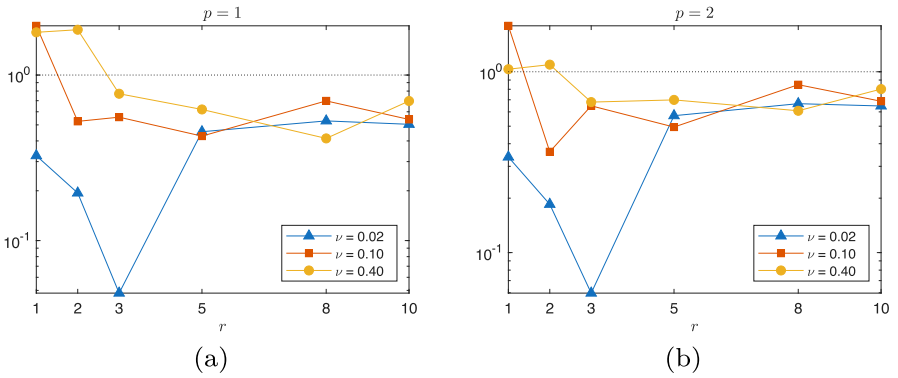


Fig. 7 Ratios $f_p(G_{\texttt{LRS}})/f_p(G_{\texttt{GoDec}})$. Values smaller than 1 indicate that the `LRS` method attains a smaller error than the `GoDec` algorithm for the given rank r and roughness level ν

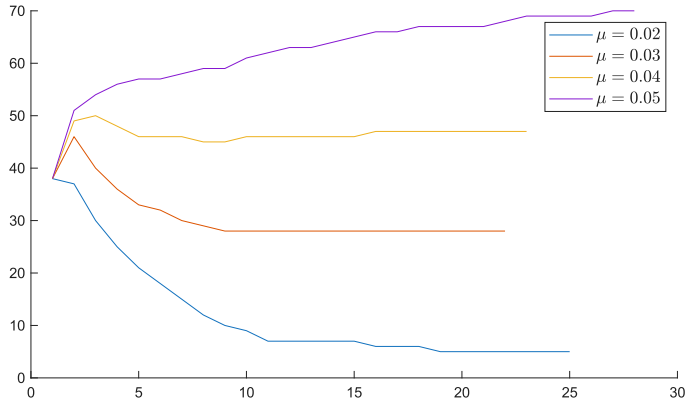


Fig. 8 Number of singular values needed for the computation of $B^{(k+1)}$ at each step of Experiment 2 when $q = 1$

Let $\hat{r}_i$ denote the number of singular values of the matrix $M^{(i)}$ defined in (7) that are larger than $1/\rho$. At iteration $k > 5$, we compute the

$$r_k = \left\lfloor \frac{1}{5} \sum_{i=k-5}^{k-1} \hat{r}_i \right\rfloor + 1$$

largest singular values, where $\lfloor \alpha \rfloor$ denotes the largest integer smaller than or equal to α. We use the MATLAB function `svds` to compute the partial SVD of $M^{(k)}$. This function implements the method described in [2]. However, we observed that, on the computer used, `svds` is faster than computing the full SVD with the MATLAB function `svd` only when

$$r_k < \tfrac{1}{10} \min\{n_1, n_2\}.$$

Therefore, whenever this inequality does not hold, we compute the full SVD regardless of the value of r_k.

Table 4 reports the computing time required to determine $B^{(k+1)}$ for problems of increasing dimension $n = n_1 = n_2$ by evaluating the partial SVD as described and the full SVD. The test

Table 4 Execution times (in seconds) for computing $B^{(k+1)}$ by evluating the partial and full SVD over 100 iterations

n	128	256	512	1024	2048
Partial SVD	0.18	0.40	1.13	5.82	40.89
Full SVD	0.14	0.69	3.91	19.49	239.44

problems were generated as discussed in Sect. 5.3 with $r = 8$, $\nu = 0.01$, and the algorithm parameters were set to $q = 1$, $K = 100$, and $\tau = 0$. We can see that computing the partial SVD as described leads to a substantial reduction in computing time for large problems.

5.5 An Application to Real Data

As a final experiment, we consider a real-world dataset. The surface displayed on top of Fig. 9 was reconstructed in [12] and represents the *Domus de Janas di Corongiu* archaeological site located in Pimentel, Sardinia, Italy.

In the three bottom rows of Fig. 9, some examples of decompositions obtained with `FLS`, `GoDec` and `LRS` are shown. We tested the three algorithms several times using a wide range of parameter values, and here we report, for each method, the reconstruction that provided the best visual result according to our inspection. For `LRS` we report a reconstruction obtained using $q = 1$. We also experimented with values $q < 1$; however, in this example, no significant improvement in the quality of the extracted glyph was achieved by considering $0 < q < 1$.

The reconstruction obtained with `LRS` is clearer than that obtained with `GoDec`, whereas the central part of the reconstruction obtained with `FLS` is better resolved. It is important to stress that the latter result is obtained by carefully fine-tuning the parameter α, whereas, in our experiments, `LRS` provided good results for a wide range of values of q. This suggests that the quality of the reconstruction obtained with `LRS` is less sensitive to the choice of the parameter.

6 Conclusion

This paper describes a novel method for extracting glyphs from a rough background. The glyph is detected by approximating the available image by a surface of low rank. This leads to a variational formulation with a regularization term based on the qth power of the ℓ_q-norm of the image for some $0 < q < 1$. Since the ℓ_q-norm for these q-values is not a norm, the minimization problem solved might not have a unique solution. Nevertheless, we show that under suitable conditions the algorithm generates a sequence of solutions with a subsequence that converges to a stationary point. Extensive numerical experiments demonstrate the advantages of the `LRS` method when compared to available methods.

Author Contributions All authors contributed equally.

Funding Open access funding provided by Università degli Studi di Cagliari within the CRUI-CARE Agreement. The authors A. A. and A. B. are members of the GNCS group of INdAM that partially funded this work with the GNCS Project 2025 (CUP E53C24001950001) "Tecniche numeriche per problemi di grandi dimensioni" and the GNCS Project 2026 "Metodi strutturati per il signal processing avanzato" (CUP E53C25002010001). A. B. is supported by the PRIN 2022 PNRR project no. P2022PMEN2 "A mathematical approach to inverse problems arising in cultural heritage preservation and dissemination" financed by the European Union - NextGenerationEU and by the Italian Ministry of University and Research (MUR).

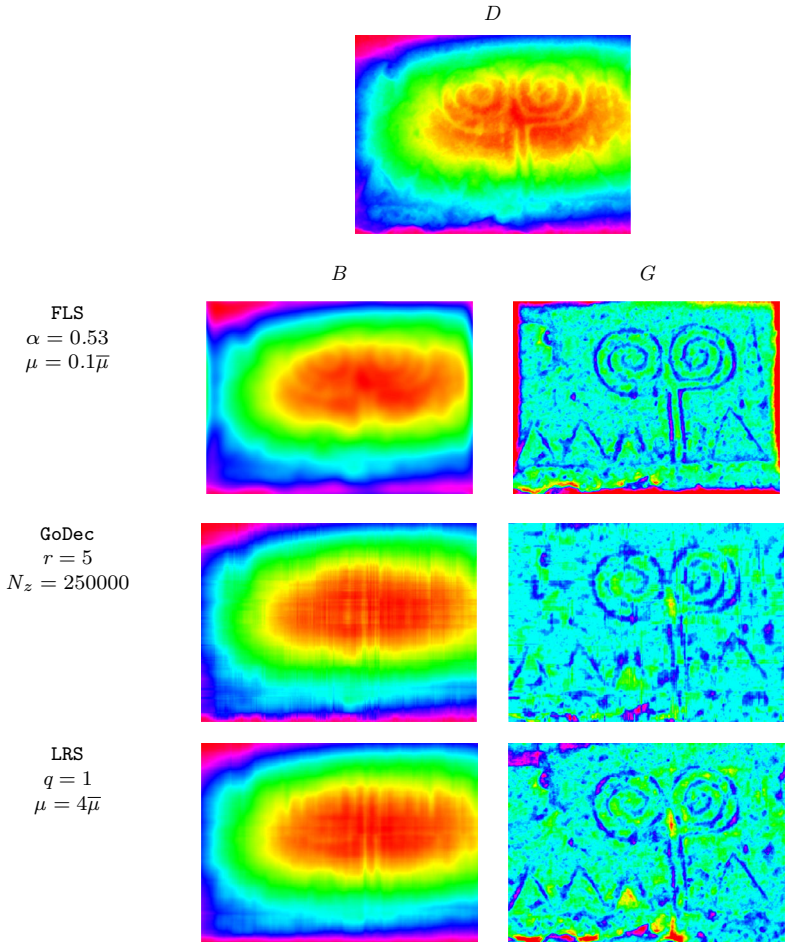


Fig. 9 *Domus de Janas di Corongiu*, Pimentel, Italy, $n_1 = 590$, $n_2 = 820$. Top: starting surface D. Left column: background component B. Right column: recovered glyph G

A. B. is partially supported by the PRIN 2022 project "Inverse Problems in the Imaging Sciences (IPIS)" (2022ANC8HL) financed by the European Union - NextGenerationEU and by the Italian Ministry of University and Research (MUR).

Data Availability Data sharing is available upon request to the authors.

Declarations

Conflict of interest The authors declare no conflict of interest during the current study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is

not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Azzarelli, A., Buccini, A.: A variational model for glyph extraction, theoretical aspects and implementation. Electron. Trans. Numer. Anal. **65**, 123–139 (2026)
2. Baglama, J., Reichel, L.: Augmented implicitly restarted Lanczos bidiagonalization methods. SIAM J. Sci. Comput. **27**, 19–42 (2005)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
4. Buccini, A., Díaz de Alba, P., Pes, F.: An alternating direction multiplier method for the inversion of FDEM data. J. Sci. Comput. **101**, 14 (2024)
5. Buccini, A., De la Cruz Cabrera, O., Donatelli, M., Martinelli, A., Reichel, L.: Large-scale regression with non-convex loss and penalty. Appl. Numer. Math. **157**, 590–601 (2020)
6. Buccini, A., Reichel, L.: An ℓ^2-ℓ^q regularization method for large discrete ill-posed problems. J. Sci. Comput. **78**(3), 1526–1549 (2019)
7. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
8. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 1–37 (2011)
9. Candès, E.J., Recht, B.: Completion nuclear norm. Found. Comput. Math. **9**, 717–772 (2009)
10. Clarke, F.H.: Optimization and Nonsmooth Analysis. SIAM, Philadelphia (1990)
11. Concas, A., Dessì, R., Fenu, C., Rodriguez, G., Vanzi, M.: Identifying the lights position in photometric stereo under unknown lighting. In: 21st International Conference on Computational Science and Its Applications (ICCSA), pp. 10–20. IEEE (2021)
12. Crabu, E., Pes, F., Rodriguez, G., Tanda, G.: Ascertaining the ideality of photometric stereo datasets under unknown lighting. Algorithms **16**(8), 375 (2023)
13. Ekeland, I., Temam, R.: Convex Analysis and Variational Problems. SIAM, Philadelphia (1999)
14. Gholami, A., Gazzola, S.: Automatic balancing parameter selection for Tikhonov-TV regularization. BIT Numer. Math. **62**(4), 1873–1898 (2022)
15. Guo, K., Liu, L., Xu, X., Xu, D., Tao, D.: GoDec+: fast and robust low-rank matrix decomposition based on maximum correntropy. IEEE Trans. Neural Netw. Learn. Syst. **29**(6), 2323–2336 (2018)
16. Hong, M., Luo, Z.Q., Razaviyayn, M.: Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. SIAM J. Optim. **26**(1), 337–364 (2016)
17. Huang, G., Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: Majorization–minimization generalized Krylov subspace methods for $\ell_p - \ell_q$ optimization applied to image restoration. BIT Numer. Math. **57**, 351–378 (2017)
18. Li, K., Wen, Y.W., Chan, R.H.: Cartoon-texture image decomposition using least squares and low-rank regularization. J. Math. Imaging Vis. **67**, 5 (2025)
19. Mordukhovich, B.S., Shao, Y.: On nonconvex subdifferential calculus in Banach spaces. J. Convex Anal. **2**(1–2), 211–227 (1995)
20. Rockafellar, R.: Characterization of the subdifferentials of convex functions. Pac. J. Math. **17**(3), 497–510 (1966)
21. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. R. Stat. Soc. B **58**, 267–288 (1996)
22. Wolke, R., Schwetlick, H.: Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons. SIAM J. Sci. Stat. Comput. **9**(5), 907–921 (1988)
23. Zhou, T., Tao, D.: Godec: randomized low-rank & sparse matrix decomposition in noisy case. Proceedings of the 28th International Conference on Machine Learning, ICML WA, USA (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.