TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

A First Order Primal-Dual Algorithm for Nonconvex TV^q Regularization

Thomas Möllenhoff, Evgeny Strekalovskiy, and Daniel Cremers

TUM-I149

A First Order Primal-Dual Algorithm for Nonconvex TV^q Regularization

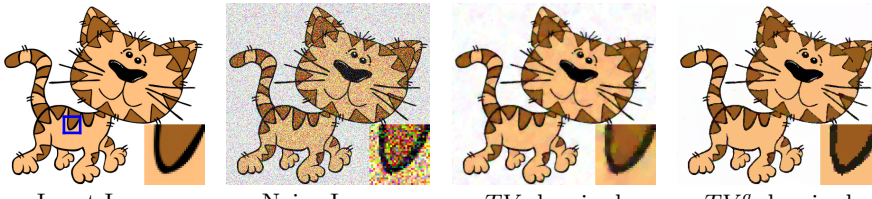
Thomas Möllenhoff, Evgeny Strekalovskiy, and Daniel Cremers

TU Munich, Germany

Abstract. We propose an efficient first order primal-dual method for solving variational problems with nonconvex regularization such as TV^q. It is based on the recent idea in [1] to reformulate an existing primal-dual algorithm for convex optimization using Moreau's identity. A systematic comparison to recent state of the art algorithms for nonconvex optimization (iteratively reweighted ℓ_1 optimization, quadratic splitting and convex relaxation methods) shows that the proposed algorithm has several advantages. Compared to iterative reweighting it does not require Lipschitz continuity or concavity of the regularizer and thus is also applicable to the case $q = 0$. Unlike the quadratic splitting approach it requires no additional hyperparameters. In contrast to convex relaxation methods it does not require a discretization of the color values and is orders of magnitudes faster and memory efficient. Numerous experiments indicate that TV^q is a well suited prior for piecewise constant images and allows to better preserve discontinuities than classical TV regularization.

1 Introduction

Regularization is of central importance in inverse problems and variational inference. In the early 1990s, *total variation* (TV) was pioneered as a discontinuity-preserving regularizer [2] and it still ranges among the most popular and versatile regularizers [3]. Nevertheless, studies on natural images indicate that the statistics of filter responses are more faithfully represented by heavy-tailed distributions giving rise to nonconvex regularizers [4, 5]. Among the most popular



Input Image (430×384) | Noisy Image (Gaussian, $\sigma = 0.3$) | TV-denoised (PSNR=22.49) | TV^q-denoised (PSNR=**23.47**)

Fig. 1. We propose a simple and efficient algorithm for solving TV^q regularized problems. The above comparison shows that for $q < 1$ (here $q = 0.5$), we are able to better preserve discontinuities and obtain higher PSNR-values than standard TV denoising.

nonconvex and discontinuity-enhancing first order regularizers are truncated linear, truncated quadratic (Mumford-Shah) and TV^q-regularizers.

While many efficient algorithms have been proposed for variational problems with convex regularization [6–8], the efficient optimization of nonconvex functionals and in particular TV^q-regularized functionals remains an open challenge.

1.1 TV^q Regularized Variational Problems

In this work we consider energy minimization problems of the following form

$$\min_{u \in X} \underbrace{\mathcal{D}(u) + \mathcal{R}(\nabla u)}_{=:E(u)} \tag{P}$$

where $\mathcal{D} : X \to \mathbb{R}$ is a convex dataterm and $\mathcal{R} : Y \to \mathbb{R}$ a nonconvex and nonsmooth regularizer.

X and Y are finite dimensional vector spaces

$$X = \left\{ u \;\middle|\; u : \Omega \to \mathbb{R}^k \right\}, \quad Y = \left\{ g \;\middle|\; g : \Omega \to \mathbb{R}^{d \times k} \right\} \tag{1}$$

where Ω is a discretized d-dimensional rectangular domain and k the number of color channels. The spaces are endowed with the standard euclidean norm $\|\cdot\|$ and inner product $\langle \cdot, \cdot \rangle$. With $\nabla : X \to Y$ we denote a linear operator mapping from X to Y describing a discretization of the gradient, e.g. by standard forward differences and $\operatorname{div} = -\nabla^*$ denotes its negative adjoint.

We also consider the linearly constrained formulation

$$\min_{u,\, g} \; \mathcal{D}(u) + \mathcal{R}(g) \quad \text{s.t.} \quad g = \nabla u, \tag{C}$$

which is equivalent to the previous formulation in the sense that every minimizer of (P) is also a minimizer of (C). The regularizer $\mathcal{R}$ is usually of the form

$$\mathcal{R}(g) = \sum_{x \in \Omega} R(g(x)) \tag{2}$$

and as a particular choice we consider the nonconvex and nonsmooth TV^q regularizer

$$R(g(x)) = \lambda \, |g(x)|^q \tag{3}$$

where $q \in [0, 1)$. With $|\cdot|$ we denote the Frobenius norm. We will stick to the convention throughout the whole paper that $\|\cdot\|$ is the norm on X or Y while $|\cdot|$ is the Euclidean norm of an element at an individual pixel location $x \in \Omega$.

The parameter $\lambda \geq 0$ describes the trade off between regularization and data fidelity. In this work we only consider a quadratic dataterm

$$\mathcal{D}(u) = \sum_{x \in \Omega} \; |u(x) - f(x)|^2 \tag{4}$$

where $f : \Omega \to \mathbb{R}^k$ is the input image and $|\cdot|$ is in this case the usual Euclidean norm on $\mathbb{R}^k$.

In the special case $q = 1$ the above model reduces to the Rudin, Osher and Fatemi (ROF) model [2]. For $q = 0$ it corresponds to the piecewise constant Mumford-Shah [9] and the Potts model [10]. In this case it is well known that finding a global minimizer amounts to solving a NP-hard optimization problem [11].

In the discrete setting it was shown by Nikolova et al. [12, Theorem 1] that minimizers of the above problem for values of $q \in (0, 1)$ are piecewise constant. This makes the above functional especially attractive for the restoration of cartoon, clip-art, text and other piecewise constant images.

1.2 Related Work

The fact that the TV^q regularizer is nonconvex, nondifferentiable and not Lipschitz continuous makes efficient optimization of related problems hard. The functional may possess many local optima and typical gradient based optimization schemes are not applicable due to nonsmoothness.

Second order methods. Hintermüller and Wu [13] tackle the TV^q problem by proposing a convergent Newton-type algorithm. While second order methods usually converge within a few iterations, they are complicated to implement, have high per iteration cost and are less amenable to parallelization. Our method significantly differs from that approach as it only consists of simple and explicit update steps that are easy to implement and parallelize.

Quadratic Splitting. Another track of algorithms are alternating minimization methods which are based on the well-known quadratic penalty method. In the classic quadratic penalty method the constrained cost function (C) is augmented by a quadratic term that penalizes constraint violation in order to arrive at an unconstrained optimization problem:

$$\min_{u,\,g} \; \mathcal{D}(u) + \mathcal{R}(g) + \frac{\tau}{2}\|g - \nabla u\|^2. \tag{5}$$

This augmented cost function is then solved for increasing values of $\tau \to \infty$. Since joint minimization over u and g is difficult, minimization is usually carried out in an alternating fashion:

$$\begin{aligned} g^{n+1} &= \arg\min_{g \in Y} \; \mathcal{R}(g) + \frac{\tau_n}{2}\|g - \nabla u^n\|^2, \\ u^{n+1} &= \arg\min_{u \in X} \; \mathcal{D}(u) + \frac{\tau_n}{2}\|\nabla u - g^{n+1}\|^2, \\ \tau_{n+1} &= \kappa\, \tau_n. \end{aligned} \tag{6}$$

The parameter $\kappa > 1$ is typically chosen between 1.05 and 2. Variations exist where the alternating minimizations are repeated several times before increasing the penalty parameter τ_n [5].

This method has appeared under many different names throughout literature such as quadratic splitting (**QS**) or quadratic decoupling/relaxation.

Krishnan et al. [5] apply it for anisotropic nonconvex TV^q-regularized inverse problems such as denoising and deblurring. For the nonconvex subproblem in g they propose to use a look-up table and give closed form solutions for $q = 1/2$ and $q = 2/3$.

A disadvantage of this approach is that it is unclear how the parameter κ should be chosen, as the solution depends heavily on that choice. Furthermore the subproblem in u requires to solve a linear system. This means that the update scheme is not fully explicit. Additionally, it is not clear how the computed solution relates to the original cost function due to the quadratic term. Our proposed method is fully explicit and does not depend on additional hyperparameters.

Iterative Reweighting. Nonconvex regularizers which are well approximated from above by a quadratic function (for instance smooth functionals) can be minimized using the iteratively reweighted least squares algorithm (IRLS). The nondifferentiable ℓ^q functions are however better approximated by an ℓ_1-norm instead of the squared ℓ_2-norm. Recently, an iterative reweighted ℓ_1 algorithm (**IRL1**) to handle linearly constrained nonconvex problems such as (P) was proposed [14].

Its update scheme consists of solving a sequence of convex ℓ_1 subproblems. In each of the subproblems one minimizes a majorizing convex approximation obtained by linearization of the initial nonconvex cost function. This assumes that the regularizer is concave and Lipschitz continuous.

Since the TV^q regularizer is *not* Lipschitz, a regularized variant is considered in [14] with a small $\varepsilon > 0$:

$$\widehat{R}(|g(x)|) = \lambda \left(|g(x)| + \varepsilon\right)^q. \tag{7}$$

In contrast to that method, our algorithm can also handle the boundary case $q = 0$ and does not require an additional ε parameter for making the penalty function Lipschitz continuous.

Convex Relaxation Methods. Another possibility of solving nonconvex problems are convex relaxation approaches. These methods try to approximate the convex envelope of the original problem. They usually provide solutions which are close to the global optimum and come with optimality bounds.

The lifting method described in [15–17] is also applicable to the TV^q problem for scalar valued images $u : \Omega \to \mathbb{R}$. We give a short description of this in Section 2.

Our proposed approach has several advantages over the convex relaxation. It has a much smaller memory footprint, orders of magnitudes lower computation times, does not require a discretization of the intensities and can be applied to vector valued images.

Nonconvex Primal-Dual Algorithm for the Mumford-Shah Functional. Strekalovskiy and Cremers [1] recently proposed an efficient algorithm for minimizing both the piecewise smooth and piecewise constant Mumford-Shah functional. By reformulating an existing primal-dual algorithm for convex optimization using Moreau's identity, they were able to extend its applicability beyond convex regularizers, and applied it to the nonconvex Mumford-Shah functional. This way they achieved promising results with real-time capacity. They considered the truncated quadratic potential term

$$R_{MS}(g(x)) = \min\left(\lambda, \alpha|g(x)|^2\right) \tag{8}$$

with $\lambda > 0$ and $\alpha > 0$. Our algorithm is based on the same idea to reformulate the primal-dual algorithm given in [8]. Thus we will give a detailed description of the approach in the next section. We note that in the limit case $q = 0$ the proposed method is identical to the algorithm from [1] for $\alpha = \infty$ in (8).

1.3 Contribution

We apply the proposed algorithm reformulation from [1] to the TV^q problem. Our contributions are as follows:

- We propose a simple and efficient algorithm for computing the arising nonconvex ℓ^q proximal mapping for arbitrary values of $q > 0$ and a concise analytic solution for the special case $q = 1/2$.
- We carry out numerical experiments to show convergence of the proposed algorithm. Furthermore we experimentally show that TV^q indeed is a well suited regularizer for the restoration of piecewise constant images.
- We provide a systematic comparison indicating that the proposed algorithm performs favorably to quadratic splitting and iteratively reweighted ℓ_1 optimization in terms of lower energies, lower computation time and higher PSNR values.
- The resulting algorithm is easy to implement and parallelize as it only consists of simple and explicit update steps.

2 Convex Relaxation for TV^q

In this section we will shortly describe how to apply the convex relaxation method from [18, 15] to the TV^q problem at hand. The method allows to convexify functionals of the form

$$\int_{\Omega\setminus S_u} h(x, u(x), \nabla u(x))\,\mathrm{d}x + \int_{S_u} d(s, u^-(s), u^+(s))\,\mathrm{d}s \tag{9}$$

where $u \in SBV(\Omega)$. The set $\Omega \subset \mathbb{R}^2$ denotes a continuous image domain and S_u the jump set of the function u where it jumps from u^- to u^+. Note that the convex relaxation approach, which is also known as functional lifthing, is only applicable to *scalar* images u. Thus we will always assume $k = 1$ in our comparisons with convex relaxation.

As mentioned at the end of Section 1.1, it is a reasonable assumption that the minimizers of the TV^q functional are piecewise constant images. Therefore we set the function h to the following

$$h(x, u(x), \nabla u(x)) = (u(x) - f(x))^2 + \begin{cases} 0 & \text{if } \nabla u(x) = 0 \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

and for the jump penalization we set

$$d(s, u^-(s), u^+(s)) = |u^-(s) - u^+(s)|^q. \tag{11}$$

For further details on how to obtain and solve the convexification of that problem using the lifting approach we refer to [17].

3 Proposed Optimization Method

3.1 Primal-Dual Algorithm [8] for Convex $\mathcal{R}$

Let us first review the primal-dual algorithm [8] in the *convex* setting since the proposed algorithm is a reformulation of that method. For this we first introduce the *Legendre-Fenchel* or *convex conjugate* [19] of a general function $f : X \to \mathbb{R}$

$$f^*(y) := \max_{x \in X} \ \langle y, x \rangle - f(x), \tag{12}$$

as well as the *proximal mapping*

$$\text{prox}_{\tau, f}(y) := \arg\min_{x \in X} \ f(x) + \frac{\|x - y\|^2}{2\tau} \tag{13}$$

for $\tau > 0$. Since for proper, convex and lower-semicontinuous functions $\mathcal{R}$ it holds that $\mathcal{R} = (\mathcal{R}^*)^*$ we can rewrite

$$\mathcal{R}(g) = \max_{q \in Y} \ \langle g, q \rangle - \mathcal{R}^*(q). \tag{14}$$

This allows us to introduce a variable splitting in the primal problem (P) by substituting (14) in (P). We arrive at the following saddle-point formulation

$$\min_{u \in X} \ \max_{q \in Y} \ \langle \nabla u, q \rangle - \mathcal{R}^*(q) + \mathcal{D}(u). \tag{PD}$$

The primal-dual algorithm aims to find a saddle-point by performing alternating gradient ascent in the dual variable q and gradient descent in the primal variable u and is given as

$$\begin{aligned} q^{n+1} &= \text{prox}_{\sigma, \mathcal{R}^*}\left(q^n + \sigma \nabla \bar{u}^n\right), \\ u^{n+1} &= \text{prox}_{\tau, \mathcal{D}}\left(u^n + \tau \operatorname{div} q^{n+1}\right), \\ \bar{u}^{n+1} &= u^n + \theta(u^{n+1} - u^n), \end{aligned} \tag{15}$$

with parameters $\sigma, \tau > 0$ and $0 \leq \theta \leq 1$. Due to the variable splitting, the proximal mappings appearing in the algorithm have a simple form. The authors of [8] prove that this algorithm converges to a saddle-point of (PD) for appropriate step sizes $\tau\sigma\|\nabla\|^2 < 1$ and $\theta = 1$.

3.2 Naive Application for Nonconvex $\mathcal{R}$

Let us now consider the naive application of algorithm (15) for the *nonconvex* regularizer

$$\mathcal{R}(g) = \sum_{x \in \Omega} R(g(x)), \quad R(g(x)) = |g(x)|^q, \qquad 0 \leq q < 1 \tag{16}$$

where $R : \mathbb{R}^{d \times k} \to \mathbb{R}$. For that we will require the Legendre-Fenchel conjugate of R. The following proposition will reveal that this conjugate is not very useful.

Proposition 1 *Let $R(g) = |g|^q$ and $0 \leq q < 1$. The Legendre-Fenchel conjugate is given by*

$$R^*(q) = \begin{cases} 0, & |q| = 0 \\ \infty, & |q| \neq 0 \end{cases} \tag{17}$$

and the biconjugate (convex envelope) $(R^*)^*$ is zero everywhere.

Proof. By definition of the Fenchel conjugate we have

$$R^*(q) = \max_{g \in \mathbb{R}^{d \times k}} \ \langle q, g \rangle - R(g) = \max_{c \in \mathbb{R}} \ c|q|^2 - |c|^q |y|^q \tag{18}$$

where we wrote $g = c\,q$ for some $c \in \mathbb{R}$ since that expression maximizes the inner product. Since for $0 \leq q < 1$ the function c grows faster than $|c|^q$ the supremum evaluates as ∞ unless $q = 0$. From the definition of $(R^*)^*$ it follows immediately that it has to be zero everywhere.

Thus the saddle-point problem (PD) is the same for all values of $0 \leq q < 1$. Furthermore, the proximal step of algorithm (15) in $\mathcal{R}^*$ just amounts to setting the iterate $q^{n+1} = 0$. Hence, the direct application of (15) with nonconvex $\mathcal{R}$ does not make much sense and does not result in a useful optimization method.

3.3 Reformulation Using Moreau's Identity

As noted in [1], the key idea is to reformulate the q-update step in (15) using Moreau's identity:

$$\operatorname{prox}_{\sigma, \mathcal{R}^*}(\widetilde{q}) = \widetilde{q} - \sigma \operatorname{prox}_{1/\sigma, \mathcal{R}}(\widetilde{q}/\sigma). \tag{19}$$

The application of Moreau's identity is valid, since we are still working in the *convex* setting. The reformulated algorithm now reads

$$\begin{aligned} \widetilde{q}^{n+1} &= q^n + \sigma \nabla \bar{u}^n, \\ q^{n+1} &= \widetilde{q}^{n+1} - \sigma \operatorname{prox}_{1/\sigma, \mathcal{R}}(\widetilde{q}^{n+1}/\sigma), \\ u^{n+1} &= \operatorname{prox}_{\tau, \mathcal{D}}(u^n + \tau \operatorname{div} q^{n+1}), \\ \bar{u}^{n+1} &= u^n + \theta(u^{n+1} - u^n), \end{aligned} \tag{20}$$

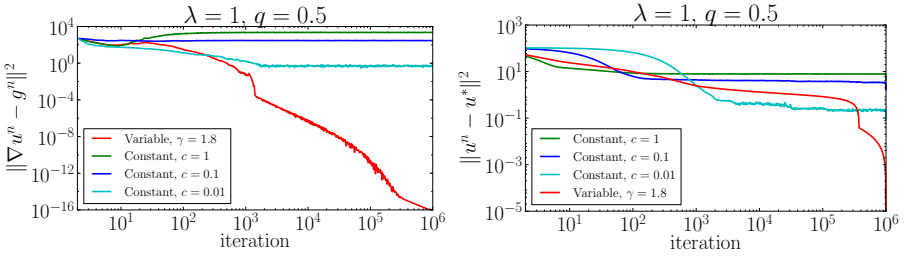


Fig. 2. Left: The primal residual for the constraint $g = \nabla u$ does not converge to zero for constant step sizes but for the variable step size scheme it does. **Right:** For each choice of step sizes we precomputed a "converged solution" $u^* = u^{10^6}$. For constant step sizes we do not observe convergence to each precomputed u^* and the error stops decreasing and starts oscillating. For the variable step size scheme the error converges.

and is completely equivalent to (15) in the convex setting in the sense that it produces exactly the same iterates. However the reformulated algorithm now does not involve $\mathcal{R}^*$ anymore, so an application to a nonconvex $\mathcal{R}$ makes more sense now.

A remaining question is the appropriate choice of step sizes σ and τ. In [1] the following choice of step sizes was suggested

$$\theta_n = 1/\sqrt{1 + 2\gamma\tau_n},\ \sigma_{n+1} = \sigma_n/\theta_n,\ \tau_{n+1} = \tau_n\theta_n, \tag{21}$$

with $\tau_0 = 1/(2d)$ and $\sigma_0 = 1/2$. This scheme works remarkably well for nonconvex regularizers, though we do not have a satisfying theoretical explanation for that yet. Furthermore, the authors of [1] show experimentally that the algorithm converges for this step size scheme and prove boundedness of the iterates.

This is the step size scheme from the accelerated primal-dual algorithm for uniformly convex functions [8, Alg. 2]. It comes with the property that the algorithm provably converges with rate $\mathcal{O}(1/n^2)$ in the convex setting. Furthermore, it is shown in [8] that $\tau_n \to 0$ and $\sigma_n \to \infty$ with rate $\mathcal{O}(1/n)$.

The parameter γ is the uniform convexity constant of the dataterm $\mathcal{D}$. Uniform convexity of $\mathcal{D}$ is defined as

$$\mathcal{D}(y) \geq \mathcal{D}(x) + \langle w, y - x\rangle + \frac{\gamma}{2}\|y - x\|^2, \quad \forall w \in \partial\mathcal{D}(x), \quad y \in X. \tag{22}$$

For the quadratic term (4) it can be quickly verified that (22) is satisfied with any $0 < \gamma \leq 2$.

3.4 Behavior of (20) for Constant Step Sizes

In our experiments the algorithm (20) for nonconvex regularizers does not converge for constant step sizes σ, τ and any overrelaxation parameter $\theta \in [0, 1]$. Furthermore it failed to find good energies. No matter how the constant step sizes were chosen the behaviour of the algorithm is oscillatory, as depicted in

Algorithm 1: Proposed Algorithm for TV^q minimization

Input: Input Image $f : \Omega \to \mathbb{R}^k$, parameters $q \geq 0$, $\lambda > 0$.
Init : $u^0 = f$, $q^0 = 0$, $\tau_0 = 1/(2d)$, $\sigma_0 = 1/2$, $\gamma = 1.8$.
for $n \geq 0$ **until** $\|u^{n+1} - u^n\|_1 / |\Omega| \leq \varepsilon$ **do**
- $\widetilde{q}^{n+1} = q^n + \sigma_n \nabla \bar{u}^n$
- $q^{n+1} =$ prox by (19) and (26) with $\widetilde{q}^{n+1}$ and σ_n
- $\widetilde{u}^{n+1} = u^n + \tau_n \operatorname{div} q^{n+1}$
- $u^{n+1} =$ prox by (24) with $\widetilde{u}^{n+1}$ and τ_n
- $\theta_n = 1/\sqrt{1 + 2\gamma\tau_n}$, $\sigma_{n+1} = \sigma_n/\theta_n$, $\tau_{n+1} = \tau_n\theta_n$
- $\bar{u}^{n+1} = u^n + \theta_n(u^{n+1} - u^n)$

Fig. 2. For the constant step sizes we picked $\tau = c/(2d)$ and $\sigma = 1/(2c)$ for various choices of $c \in \{10^0, 10^{-1}, 10^{-2}\}$. As suggested by Fig. 2 it seems that smaller step sizes lead to better results. So the step size scheme suggested in [1] where $\sigma_n \to \infty$, $\tau_n \to 0$, $\theta_n \to 1$ is a natural choice. We observed the best performance with the choice $\gamma = 1.8$ and used it throughout all our experiments.

3.5 Interpretation of the Reformulated Algorithm (20)

Let us further define an auxiliary variable g for the prox-result in the update step for q^{n+1} in (20). This allows us to rewrite the algorithm in the following way:

$$\begin{aligned} g^{n+1} &= \operatorname{prox}_{1/\sigma,\mathcal{R}}(\nabla \bar{u}^n + (1/\sigma)q^n), \\ q^{n+1} &= q^n + \sigma\left(\nabla \bar{u}^n - g^{n+1}\right), \\ u^{n+1} &= \operatorname{prox}_{\tau,\mathcal{D}}(u^n + \tau \operatorname{div} q^{n+1}), \\ \bar{u}^{n+1} &= u^n + \theta(u^{n+1} - u^n). \end{aligned} \tag{23}$$

Let us consider the saddle-point problem that is obtained by considering the constrained primal problem (C) and introducing a Lagrange multiplier q for the constraint. The resulting saddle-point formulation is given as

$$\min_{u,\,g} \max_{q} \; \mathcal{D}(u) + \mathcal{R}(g) + \langle q, \nabla u - g\rangle. \tag{PD*}$$

The update step in g^{n+1} in (23) can be interpreted as a gradient descent step with step size $1/\sigma$ on the saddle-point function (PD*) in the variable g, except that instead of the usual starting point g^n we use $\nabla \bar{u}^n$ which should be equal to g^n in the limit $n \to \infty$. The update step in q^{n+1} in (23) is a gradient ascent step for the Lagrange multiplier q with step size σ. Finally, the update step in u^{n+1} is again a gradient descent step in the variable u.

3.6 Evaluation of the Proximal Mappings

What is left in order to apply the algorithm (20) is to calculate the proximal mappings involving $\mathcal{D}$ and $\mathcal{R}$. Since there is no coupling between the individual

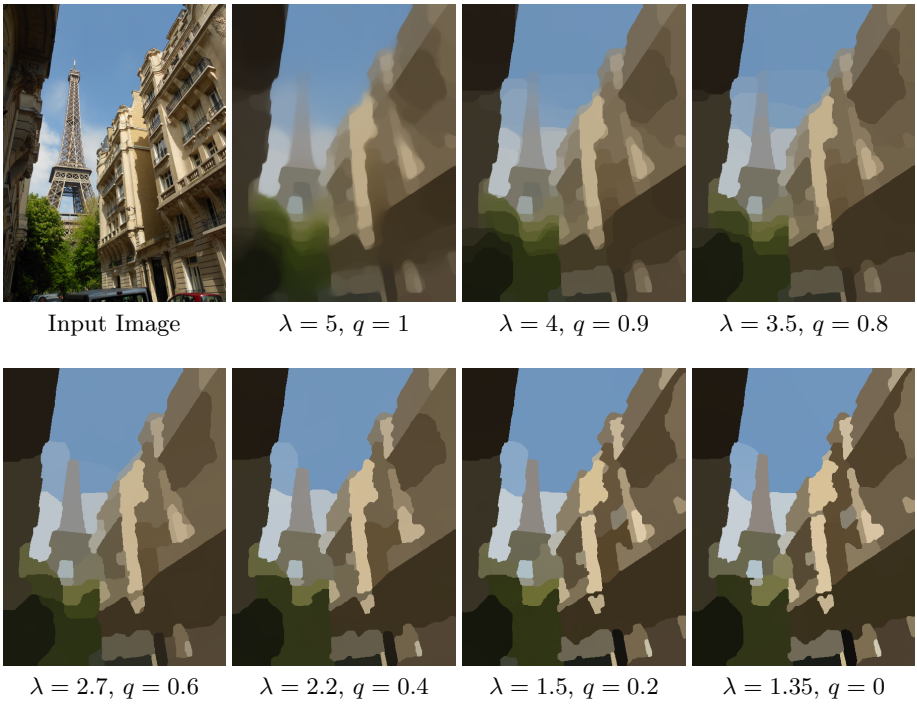


Fig. 3. Effect of the parameter q illustrated on a color image. Values of $q < 1$ lead to piecewise constant results and smaller values of q lead to higher contrast between the regions. λ was manually adjusted so adjacent images have the highest visual similarity.

pixels anymore, they can be evaluated pointwise for every $x \in \Omega$. For notational convenience we introduce the functions $D : \mathbb{R}^k \to \mathbb{R}$ and $R : \mathbb{R}^{d \times k} \to \mathbb{R}$ which correspond to the pointwise terms.

The proximal mapping for D does not pose any difficulties. By a short calculation (see e.g. [8]) it can be obtained that the minimizer $\widehat{u} \in \mathbb{R}^k$ of the optimization problem

$$\operatorname{prox}_{\tau, D}(\widetilde{u}) = \operatorname*{arg\,min}_{u \in \mathbb{R}^k} \ |u - f|^2 + \frac{|u - \widetilde{u}|^2}{2\tau} \tag{24}$$

is given by

$$\widehat{u} = \frac{\widetilde{u} + 2\tau f}{1 + 2\tau} \tag{25}$$

with $\widetilde{u} \in \mathbb{R}^k$. $f \in \mathbb{R}^k$ is the input image evaluated at the according point $x \in \Omega$.

The evaluation of the proximal mapping in R is more involved since it cannot be carried out in closed form for the generic case $0 \leq q < 1$. It requires finding the solution to the following minimization problem:

$$\operatorname{prox}_{\tau, R}(\widetilde{g}) = \operatorname*{arg\,min}_{g \in \mathbb{R}^{d \times k}} \ \frac{|g - \widetilde{g}|^2}{2\tau} + \lambda |g|^q. \tag{26}$$

Input (256 × 229) | Noisy ($\sigma = 0.5$) PSNR=6.0 | $q = 0.2$, $\lambda = 1$ a) PSNR=14.3 | $q = 0.2$, $\lambda = 1.6$ b) PSNR=16.5 | $q = 0.8$, $\lambda = 1.1$ c) PSNR=18.4

Fig. 4. The effect of the parameter q for image denoising. **a)**, **b)** While smaller values of q lead to sharp boundaries and clearer regions, large noise outliers are not being removed since big jumps get penalized less. Higher values of λ are required in order to remove them, which leads to the removal of detailed structures. **c)** We found values around $q \approx 0.8$ to yield the overall highest PSNR values as it describes a good trade-off.

The case $q = 0$. An important special case is $q = 0$, which corresponds to Potts regularization. In this case the minimization (26) can actually be solved explicitly as described in [1]:

$$\operatorname{prox}_{\tau,R}(\widetilde{g}) = \begin{cases} 0 & \text{if } |\widetilde{g}| \leq \sqrt{2\tau\lambda} \\ \widetilde{g} & \text{otherwise.} \end{cases} \tag{27}$$

The general case $0 < q < 1$. For the general case we first note that the evaluation of the proximal operator (26) can be reduced to a problem in a single scalar variable.

Proposition 2 *Given $\widetilde{g} \in \mathbb{R}^{d\times k}$, $\tau > 0$, $q \in (0,1)$ and $\lambda > 0$, the solution of the proximal operator*

$$\operatorname{prox}_{\tau,R}(\widetilde{g}) = \operatorname*{arg\,min}_{g\in\mathbb{R}^{d\times k}} \frac{|g-\widetilde{g}|^2}{2\tau} + \lambda|g|^q$$

has the form $\widehat{g} = t\widetilde{g}$ for some real $t \geq 0$.

Proof. A proof is given in the appendix of [20].

Since we now know that the optimal solution is a scalar multiple of $\widetilde{g}$ we substitute $g = t\widetilde{g}$ in (26) and get the following new scalar problem

$$\operatorname*{arg\,min}_{t\geq 0} \frac{(t-1)^2}{2} + \alpha t^q \tag{28}$$

for $\alpha = \tau\lambda|\widetilde{g}|^{q-2} \geq 0$. Thus, evaluating the proximal operator (26) reduces to solving the above problem (28) for $t \geq 0$.

The minimization problem (28) can be solved in closed form for certain values of q such as $1/2$ or $3/4$ as described in [5]. We provide a more concise analytic

Input Image (200×182) | Noisy, $\sigma = 0.3$ (PSNR=10.5) | $u^0 = f$ (PSNR=19.8) | Random u^0 (PSNR=20.7) | $u^0 = 0$ (PSNR=20.8)

Fig. 5. We show the effect of using different initializations ($\lambda = 0.5$, $q = 0.5$). Since the optimization problem is highly nonconvex it might have many local minima. If we initialize the algorithm with the degraded image, the algorithm often gets trapped in a bad local minimum where the noise outliers are not getting removed. This effect becomes worse for smaller values of q and higher noise variances. The results produced by using a random or zero initialization are almost identical.

solution for the special case $1/2$ and an efficient algorithm based on Newton's method for the general case in the appendix.

The final nonconvex primal-dual algorithm for TV^q denoising is summarized in Algorithm 1. We stop the iterations when there is no significant change in the solution anymore, which we observed to be the case when $\|u_n - u_{n-1}\|_1/|\Omega| \leq \varepsilon$. In practice we set $\varepsilon = 5 \cdot 10^{-5}$.

4 Numerical Experiments

We implemented all considered algorithms using CUDA to run on GPUs. We used `double` precision for the convergence experiments and `float` precision for all other experiments. All numerical experiments were carried out on a GTX680 graphics card. We used cuFFT [1] for the FFT calculations required by the quadratic splitting approach.

4.1 Effect of TV^q Regularization

In Fig. 3 we show the effect of the parameter q on a natural image. We show the results produced by our method for various values of q. The regularization parameter λ was adjusted so that two adjacent images in Fig. 3 have the lowest Euclidean distance.

Values of $q < 1$ lead to piecewise constant approximations and for smaller values of q we observe higher contrast between the regions. That is because for smaller values of q, bigger jumps are penalized less and less. In the border case $q = 0$ all jumps are penalized equally. Note that the proposed algorithm produces consistent results in a sense that smaller values of q systematically lead to a higher contrast between the regions.

[1] https://developer.nvidia.com/cufft

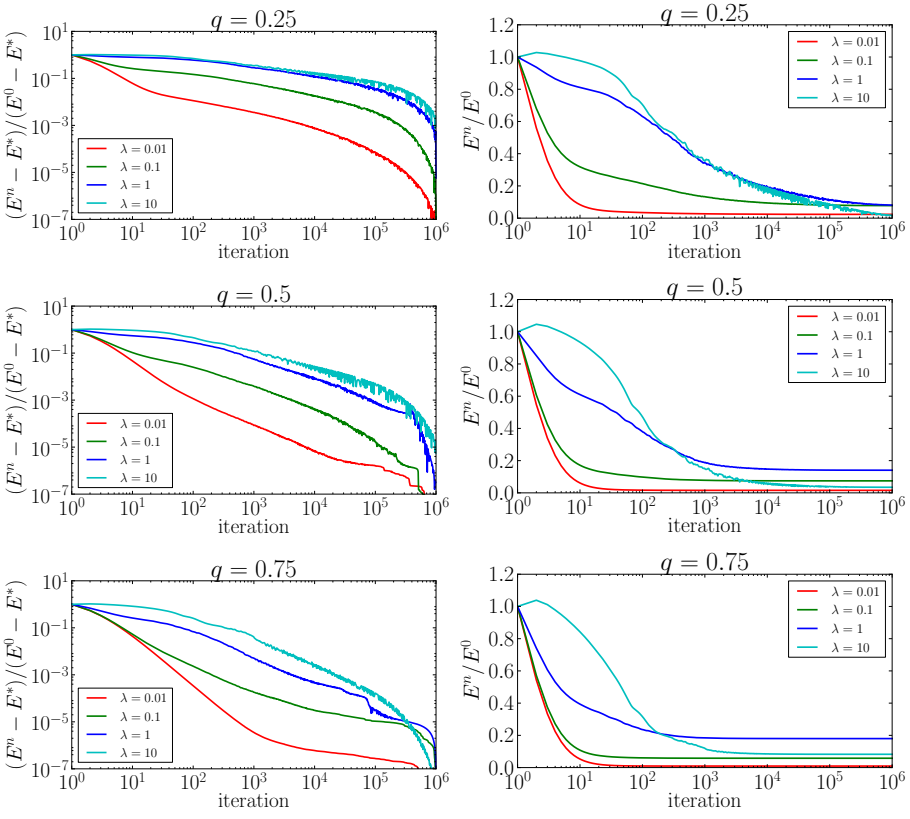


Fig. 6. Experimental convergence of the proposed algorithm on the 256×256 lena image. We observe convergence for both the normalized energy $(E^n - E^*)/(E^0 - E^*)$ and the normalized energy E^n/E^0 for various values of q and λ.

In Fig. 4 we illustrate how the denoising performance of the algorithm depends on the parameter q. While smaller values of q lead to desirable sharper boundaries and higher contrast, strong noise outliers do not get removed anymore due to the lower penalization of large jumps. Finding the correct value of q means finding a good trade off and values of $q \approx 0.75$ lead to the highest PSNR.

4.2 Dependance on the Initialization

Since the optimization problem is nonconvex we show the dependance of the solution on the initialization in Fig. 5. If we initialize the algorithm with a noisy input image we observe overall higher energies and a more noisy final solution. Initializing the algorithm with a constant image leads to the best results. Although the initial problem is nonconvex, the proposed algorithm turns out to be surprisingly robust w.r.t initial values.

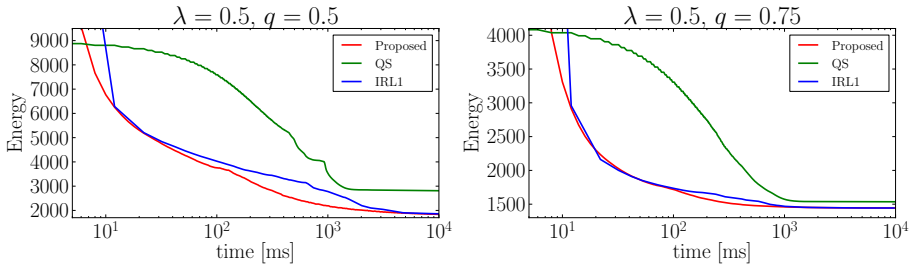


Fig. 7. We show the energy decrease over time for the image from Fig. 8 and two different values of q. Our proposed algorithm minimizes the energy functional faster than the iterative reweighted ℓ_1 and quadratic splitting method. The quadratic splitting method converges to a significantly higher energy, especially for lower values of q.

4.3 Convergence of the Energy

Since there is no theoretical proof of convergence yet, we validated the convergence experimentally. We precomputed a $u^* = u^{10^6}$ as an approximation to the converged solution. It can be seen in Fig. 7 that the normalized energies $(E(u^n) - E(u^*))/(E(u^0) - E(u^*))$ and $E(u^n)/E(u^0)$ decrease and converge.

Furthermore, we compare the energy decrease of the proposed method over time to the iterative reweighted ℓ_1 (IRL1) and quadratic splitting (QS) method in Fig. 7. For the iterative reweighting method we chose $\varepsilon = 10^{-6}$ as a regularization parameter to make the TV^q function Lipschitz continuous. For the iterative reweighting method we solve the inner convex optimization problem using the same accelerated primal-dual algorithm.

Furthermore we chose $\kappa = 1.05$ in the quadratic splitting method. We initialized all algorithms with the input image ($u^0 = f$).

For $q = 0.75$ the proposed algorithm finds the lowest energies significantly faster than the other methods. For smaller values of q such as 0.5 our method performs comparably to iterative reweighting.

The final energy found by the proposed method and iterative reweighting are almost identical, while the quadratic splitting method converges to a significantly higher energy.

4.4 Comparison to Other Methods

In Fig. 8 we compute a piecewise constant approximation of the input image using the different algorithms. The figure shows that for larger values of q all methods yield visually similar results, but for small values of q such as 0.05 the proposed method finds solutions which are closest to the convex relaxation.

We note that the convex relaxation method took over 10^5 iterations to produce a piecewise constant result, which took over 10 minutes to compute on the GPU. The proposed method yields a result of similar quality in a few hundred milliseconds.

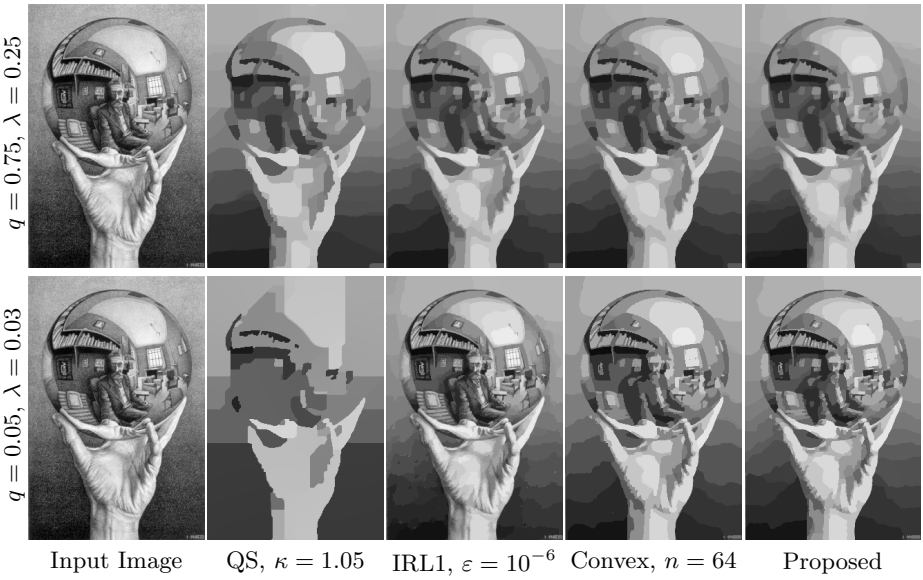


Fig. 8. Piecewise constant approximation of Escher's "Hand with Reflecting Sphere, 1935" by TV^q regularization using the different algorithms. **Top row:** For values of q that are close to 1 all methods yield comparable results . **Bottom row:** Unlike the other approaches the proposed method gives comparable results to the convex relaxation method, while requiring orders of magnitude less runtime and memory.

4.5 Denoising of Piecewise Constant Images

In Fig. 9 the denoising performance of the different algorithms on a piecewise constant image is shown. We chose $q = 0.8$ and the hyperparameter λ was chosen in order to obtain the highest PSNR values. We let the methods run until the according convergence criterion is met.

Our algorithm and iteratively reweighted ℓ_1 optimization find slightly higher PSNR values than the quadratic splitting method. In terms of runtime the proposed approach outperforms the other algorithms.

5 Conclusion

Following the recent idea from [1] to reformulate a primal-dual algorithm from convex optimization using Moreau's identity, we proposed an efficient method for solving TV^q regularized problems. We gave experimental evidence that the proposed algorithm converges for the suggested variable step size scheme and showed that it does not converge for constant step sizes. Further experiments suggest that our method performs comparably to state-of-the-art competitors such as iteratively reweighted ℓ_1 optimization.

Furthermore we proposed an efficient method for solving the occurring non-convex proximal mapping and gave a concise closed form solution for $q = 1/2$.

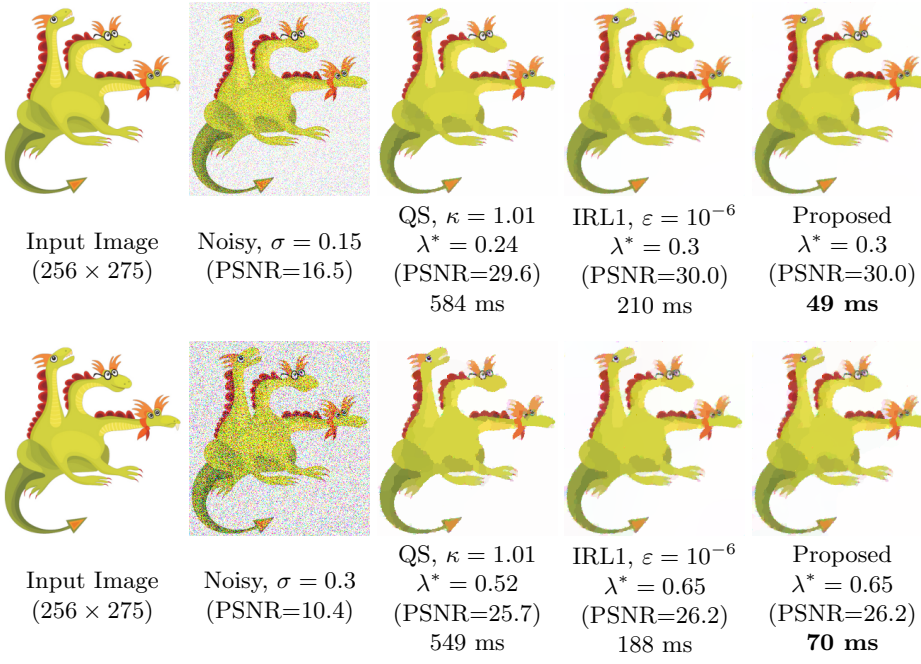


Fig. 9. Denoising of a piecewise constant image with TV^q regularization (here $q = 0.8$). The λ parameter was manually chosen for highest PSNR. The proposed algorithm finds comparable PSNR values to the iterative reweighting method and outperforms the quadratic splitting approach. It is significantly faster than the other methods.

Our proposed method has several favorable properties. Its simple structure makes it easy to implement and parallelize. Compared to the iterative reweighting method it does not require the regularizer to be Lipschitz continuous or concave, so we do not require a smoothing of the cost function and our algorithm is also applicable to the border case $q = 0$.

In contrast to the quadratic splitting method we do not require a heuristical parameter κ and we have shown that our algorithm finds solutions in terms of much higher quality.

Unlike the convex relaxation approach we do not require a discretization of the intensity values and our approach has orders of magnitude lower memory requirements and runtimes. Furthermore it is applicable to vector valued functions such as color images.

We believe that the proposed algorithm will be of great utility for efficiently solving TV^q regularized problems. For future work we want to study more theoretical properties of the algorithm such as convergence, optimality and the appropriate choice of step sizes. Extensions to general inverse problems, problems with non-quadratic dataterms and to more advanced regularizers are also left for future work.

References

1. Strekalovskiy, E., Cremers, D.: Real-Time Minimization of the Piecewise Smooth Mumford-Shah Functional. Technical Report TUM-I146, TU Munich (2014)
2. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60** (1992) 259–268
3. Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T.: An introduction to total variation for image analysis. In: Theoretical Foundations and Numerical Methods for Sparse Recovery. De Gruyter (2010)
4. Huang, J., Mumford, D.: Statistics of natural images and models. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR). (1999)
5. D. Krishnan and R. Fergus: Fast Image Deconvolution using Hyper-Laplacian Priors. In: Proc. Neural Information Processing Systems. (2009) 1033–1041
6. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical Report 08-34, UCLA Cam Report (2008)
7. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In: IEEE Int. Conf. on Computer Vision (ICCV), Kyoto, Japan (2009)
8. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40** (2011) 120–145
9. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42** (1989) 577–685
10. Potts, R.B.: Some generalized order-disorder transformations. Proc. Camb. Phil. Soc. **48** (1952) 106–109
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Patt. Anal. and Mach. Intell. **23**(11) (2001) 1222–1239
12. Nikolova, M., Ng, M.K., Zhang, S., Ching, W.K.: Efficient Reconstruction of Piecewise Constant Images Using Nonsmooth Nonconvex Minimization. SIAM J. Imaging Sciences **1**(1) (2008) 2–25
13. Hintermüller, M., Wu, T.: Nonconvex TV^q-models in image restoration: Analysis and a trust-region regularization–based superlinearly convergent solver. SIAM Journal on Imaging Sciences **6**(3) (2013) 1385–1415
14. Ochs, P., Dosovitskiy, A., Pock, T., Brox, T.: An iterated L1 Algorithm for Non-smooth Non-convex Optimization in Computer Vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2013)
15. Chambolle, A., Cremers, D., Pock, T.: A convex approach to minimal partitions. SIAM Journal on Imaging Sciences **5**(4) (2012) 1113–1158
16. Strekalovskiy, E., Cremers, D.: Total variation for cyclic structures: Convex relaxation and efficient minimization. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR). (June 2011)
17. Cremers, D., Strekalovskiy, E.: Total cyclic variation and generalizations. Journal of mathematical imaging and vision **47**(3) (2013) 258–277
18. Alberti, G., Bouchitté, G., Maso, G.D.: The calibration method for the mumford-shah functional and free-discontinuity problems. Calc. Var. Partial Dif. **3**(16) (2003) 299–333
19. Rockafellar, R.T.: Convex Analysis. Princeton University Press (1996)
20. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse Iterative Closest Point. Computer Graphics Forum (Symposium on Geometry Processing) **32**(5) (2013) 1–11
21. McKelvey, J.P.: Simple transcendental expressions for the roots of cubic equations. Amer. J. Phys. **52**(3) (March 1984) 269–270

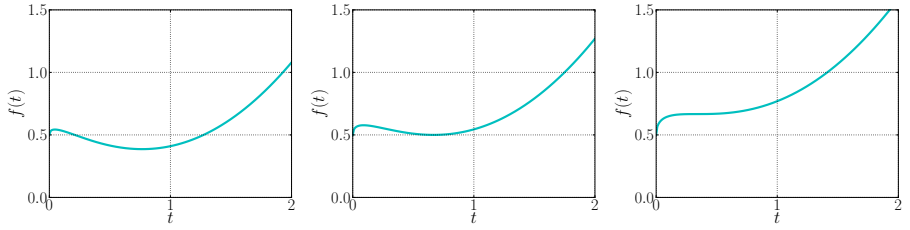


Fig. 10. The cost function (29) for three different increasing values of α. **From left to right:** $\alpha \approx 0.41 < \frac{2\sqrt{6}}{9}$, $\alpha = \frac{2\sqrt{6}}{9} \approx 0.54$, $\alpha = \frac{4}{3\sqrt{3}} \approx 0.77$. It can be seen that the desired stationary point is also the global minimum for $\alpha < \frac{2\sqrt{6}}{9}$ (left). In the equality case, the value at the stationary point is the same as the boundary value (center). For $\alpha > \frac{4}{3\sqrt{3}}$ the cost function is increasing (right).

A Evaluating the ℓ^q Proximal Mapping

As shown in the paper, evaluating the proximal operator (26) requires solving the scalar ℓ_q problem (28). Here we give detailed information on how to solve this problem either in closed form for $q = 1/2$ or with Newton's method.

A.1 Closed form solution for $q = 1/2$

We will first focus on the special case $q = 1/2$. The problem has the form

$$\arg\min_{t \geq 0} \quad \frac{(t-1)^2}{2} + \alpha\, t^{1/2} \tag{29}$$

with $\alpha = \lambda\tau|\widetilde{g}|^{-1.5} \geq 0$.

Setting the derivative of the cost function (29) with respect to t to 0 and substituting $t = s^2$ we arrive at the following cubic equation:

$$s^3 - s + \frac{\alpha}{2} = 0 \tag{30}$$

Following the work of [21] we arrive at the following closed form expression for the root which corresponds to the minimum of (29):

$$\widehat{s} = \frac{2}{\sqrt{3}} \sin\Big(\frac{1}{3}\Big(\arccos\Big(\frac{3\sqrt{3}}{4}\alpha\Big) + \frac{\pi}{2}\Big)\Big) \tag{31}$$

Fig. 10 shows that for some values of α, the value at the boundary is the optimal value. It can be shown that this holds for all α satisfying condition (32).

$$\alpha > \frac{2\sqrt{6}}{9} \tag{32}$$

In that case, we simply set $\widehat{t} = 0$. Otherwise we find the root $\widehat{s}$ of (30) that corresponds to the local minimum using formula (31) and set $\widehat{t} = \widehat{s}^2$.

Algorithm 2: Proposed algorithm for solving the ℓ^q proximal operator.

Input: Parameters $\widetilde{g} \in \mathbb{R}^{d\times k}$, $\tau > 0$, $\lambda > 0$ and $q \in (0,1)$, precision $\varepsilon > 0$
Output: Minimizer $\widehat{g} \in \mathbb{R}^{d\times k}$ of (26)
if $|\widetilde{g}| > 0$ **then**
  $\alpha = \tau\lambda|\widetilde{g}|^{q-2}$
  if α *satisfies* (34) **then**
    $\widehat{t} = 0$
  else
    // Solve for optimum using Newton's method.
    $t_0 = 1$
    for $k \geq 1$ **until** $f'(t_k)/f''(t_k) < \varepsilon$ **do**
      $t_k = t_{k-1} - f'(t_k)/f''(t_k)$
    $\widehat{t} = t_k$
  $\widehat{g} = \widehat{t}\,\widetilde{g}$
else
  $\widehat{g} = 0$ // In the case $\widetilde{g} = 0$ we can just set $\widehat{g} = 0$.

A.2 Newton's method for general $0 < q < 1$

For general values of q, we solve the scalar ℓ_q problem

$$\arg\min_{t\geq 0} \quad \frac{(t-1)^2}{2} + \alpha t^q \tag{33}$$

using Newton's method. A straightforward calculation reveals that for all α satisfying the condition

$$\alpha > \frac{1}{2-q}\left(2\,\frac{1-q}{2-q}\right)^{1-q} \tag{34}$$

the boundary value is lower than the value at the local minimum, so we set $\widehat{t} = 0$ if (34) is satisfied and otherwise we use Newton's method.

Note that for $\alpha = 0$ the optimal point is at $\widehat{t} = 1$, and for $\alpha > 0$ we have $\widehat{t} < 1$. So we pick the starting point for Newton's method $t_0 = 1$. We perform the iteration

$$t_{k+1} = t_k - f'(t_k)/f''(t_k)$$

where $f(t) = \frac{(t-1)^2}{2} + \alpha\, t^q$ and f' and f'' denote the first and second derivatives of f. It can be easily shown that the function f' is convex and increasing on the closed interval $\left[\widehat{t}, 1\right]$, so Newton's method converges to the minimum.

The final algorithm to evaluate the proximal operator (26) using Newton's method is given as Algorithm 2. In practice, this algorithm converges to the minimum within a 2–3 iterations even when ε is set to machine precision.