An iterated ℓ_1 Algorithm for Non-smooth Non-convex Optimization in Computer Vision

Peter Ochs[1], Alexey Dosovitskiy[1], Thomas Brox[1], and Thomas Pock[2]

[1] University of Freiburg, Germany
{ochs,dosovits,brox}@cs.uni-freiburg.de

[2] Graz University of Technology, Austria
pock@icg.tugraz.at

Abstract

Natural image statistics indicate that we should use non-convex norms for most regularization tasks in image processing and computer vision. Still, they are rarely used in practice due to the challenge to optimize them. Recently, iteratively reweighed ℓ_1 minimization has been proposed as a way to tackle a class of non-convex functions by solving a sequence of convex ℓ_2-ℓ_1 problems. Here we extend the problem class to linearly constrained optimization of a Lipschitz continuous function, which is the sum of a convex function and a function being concave and increasing on the non-negative orthant (possibly non-convex and non-concave on the whole space). This allows to apply the algorithm to many computer vision tasks. We show the effect of non-convex regularizers on image denoising, deconvolution, optical flow, and depth map fusion. Non-convexity is particularly interesting in combination with total generalized variation and learned image priors. Efficient optimization is made possible by some important properties that are shown to hold.

1. Introduction

Modeling and optimization with variational methods in computer vision are like antagonists on a balance scale. A major modification of a variational approach always requires developing suitable numerical algorithms.

About two decades ago, people started to replace quadratic regularization terms by non-smooth ℓ_1 terms [24], in order to improve the edge-preserving ability of the models. Although, initially, algorithms were very slow, now, state-of-the-art convex optimization techniques show comparable efficiency to quadratic problems [8].

The development in the non-convex world turns out to be much more difficult. Indeed, in a SIAM review in 1993, R. Rockafellar pointed out that: "The great watershed in optimization is not between linearity and non-linearity, but convexity and non-convexity". This statement has been enforced by deriving the worst-case complexity bounds for general non-convex problems in [17] and makes it seemingly hopeless to find efficient algorithms in the non-convex case. However, there exist particular instances that still allow for efficient numerical algorithms.

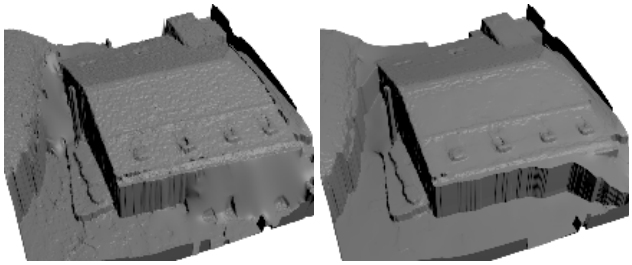

Figure 1. The depth map fusion result of a stack of depth maps is shown as a 3D rendering. Total generalized variation regularization used for the fusion has the property to favor piecewise affine functions like the roof or the street. However, there is a trade-off between affine pieces and discontinuities. For convex ℓ_1-norm regularization (**left**) this trade-off is rather sensible. This paper enables the optimization of non-convex norms (**right**) which emphasize the model properties and perform better for many computer vision tasks.

In this paper, we show that a certain class of linearly constrained convex plus concave (only on the non-negative orthant) optimization problems are particularly suitable for computer vision problems and can be efficiently minimized using state-of-the-art algorithms from convex optimization.

- We show how this class of problems can be efficiently optimized by minimizing a sequence of convex problems.

- We prove that the proposed algorithm monotonically decreases the function value of the original problem, which makes the algorithm an efficient tool for practical applications. Moreover, under slightly restricted conditions, we show existence of accumulation points and, that each accumulation point is a stationary point.

- In computer vision examples like image denoising, deconvolution, optical flow, and depth map fusion, we demonstrate that non-convex models consistently outperform their convex counterparts.

2. Related work

Since the seminal works of Geman and Geman [13], Blake and Zissermann [5], and Mumford and Shah [16] on image restoration, the application of non-convex potential functions in variational approaches for computer vision problems has become a standard paradigm. The non-convexity can be motivated and justified from different viewpoints, including robust statistics [4], nonlinear partial differential equations [20], and natural image statistics [14].

Since then, numerous works demonstrated through experiments [4, 23], that non-convex potential functions are the right choice. However, their usage makes it very hard to find a good minimizer. Early approaches are based on annealing-type schemes [13] and continuation methods such as the graduated non-convexity (GNC) algorithm [5]. However, these approaches are very slow and their results heavily depend on the initial guess. A first breakthrough was achieved by Geman and Reynolds [12]. They rewrote the (smooth) non-convex potential function as the infimum over a family of quadratic functions. This transformation suggests an algorithmic scheme that solves a sequence of quadratic problems, leading to the so-called iteratively reweighted least squares (IRLS) algorithm. This algorithm quickly became a standard solver and hence, it has been extended and studied in many works, see e.g. [26, 19, 10].

The IRLS algorithm can only be applied if the non-convex function can be well approximated from above with quadratic functions. This does not cover the non-convex ℓ_p pseudo-norms, $p \in (0, 1)$, which are non-differentiable at zero. Candes et al. [7] tackled this problem by the so-called iteratively reweighted ℓ_1 (IRL1) algorithm. It solves a sequence of non-smooth ℓ_1 problems and hence can be seen as non-smooth counterpart to the IRLS algorithm. Originally, the IRL1 algorithm was proposed to improve the sparsity properties in ℓ_1 regularized compressed sensing problems, but it turns out that this algorithm is also useful for computer vision applications.

First convergence results for the IRL1 algorithm have been obtained by Chen et al. in [9] for a class of non-convex ℓ_2-ℓ_p problems used in sparse recovery. In particular, they show that the method monotonically decreases the energy of the non-convex problem. Unfortunately, the class of problems they considered is not suitable for typical computer vision problems, due to the absence of a linear operator that is needed in order to represent spatial regularization terms.

Another track of algorithms considering non-convex objectives is the difference of convex functions (DC) programming [2]. The general DC algorithm (DCA) alternates between minimizing the difference of the convex dual functions and the difference of the convex functions. In the practical DCA convex programs obtained by linearizing one of the two functions are solved alternately. Applying DC programming to the function class of the IRL1 algorithm requires an "unnatural" splitting of the objective function. It makes the optimization hard as emerging proximity operators are difficult to solve in closed form.

Therefore, we focus on generalizing the IRL1 algorithm, present a thorough analysis of this new optimization framework, and make it applicable to computer vision problems.

3. A linearly constrained non-smooth and non-convex optimization problem

In this paper we study a wide class of optimization problems, which include ℓ_2-ℓ_p and ℓ_1-ℓ_p problems with $0 < p < 1$. These are highly interesting for many computer vision applications as will be demonstrated in Section 4. The model we consider is a linearly constrained minimization problem on a finite dimensional Hilbert space $\mathcal{H}$ of the form

$$\min_{x \in \mathcal{H}} F(x), \quad \text{s.t. } Ax = b, \tag{1}$$

with $F \colon \mathcal{H} \to \mathbb{R}$ being a sum of two Lipschitz continuous terms

$$F(x) := F_1(x) + F_2(|x|).$$

In addition we suppose that F is bounded from below, $F_1 \colon \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is proper convex and $F_2 \colon \mathcal{H}_+ \to \mathbb{R}$ is concave and increasing. Here, $\mathcal{H}_+$ denotes the non-negative orthant of the space $\mathcal{H}$; increasingness and the absolute value $|x|$ are to be understood coordinate-wise. The linear constraint $Ax = b$ is given by a linear operator $A \colon \mathcal{H} \to \mathcal{H}_1$, mapping $\mathcal{H}$ into another finite dimensional Hilbert space $\mathcal{H}_1$, and a vector $b \in \mathcal{H}_1$.

As a special case, we obtain the formulation [9]

$$F_1(x) = \|Tx - g\|_2^2, \quad \text{and} \quad F_2(|x|) = \lambda \|x\|_{\varepsilon,p}^p,$$

where $\|x\|_{\varepsilon,p}^p = \sum_i (|x_i| + \varepsilon)^p$ is a non-convex norm for $0 < p < 1$, $\lambda \in \mathbb{R}_+$, T is a linear operator, and g is a vector to be approximated. This kind of variational approach comes from compressed sensing and is related but not general enough for computer vision tasks. In [9] an iteratively reweighted ℓ_1 minimization algorithm is proposed to tackle this problem. In the next subsections, we propose a generalized version of the algorithm, followed by a convergence analysis, which supplies important insights for the final implementation.

3.1. Iteratively reweighted ℓ_1 minimization

For solving the optimization problem (1) we propose the following algorithm:

$$\begin{aligned} x^{k+1} &= \arg\min_{Ax=b} F^k(x) \\ &:= \arg\min_{Ax=b} F_1(x) + \|w^k \cdot x\|_1, \end{aligned} \tag{2}$$

where $w^k \cdot x$ is the coordinate-wise product of the vectors w^k and x, and w^k is any vector satisfying

$$w^k \in \overline{\partial} F_2(|x^k|), \tag{3}$$

where $\overline{\partial} F_2$ denotes the superdifferential[1] of the concave function F_2. We note that since F_2 is increasing, the vector w^k has non-negative components.

The algorithm proceeds by iteratively solving ℓ_1 problems which approximate the original problem. As F_1 is convex, (2) is a linearly constrained non-smooth convex optimization problem, which can be solved efficiently [8, 3, 18]. For more details on the algorithmic issue, see Section 4.

3.2. Convergence analysis

Our analysis proceeds in much the same way as [9]:

1. Show that the sequence $(F(x^k))$ is monotonically decreasing and converging.
2. Under additional constraints show the existence of an accumulation point of the sequence (x^k).
3. Under additional constraints show that any accumulation point of the sequence (x^k) is a stationary point of (1).

Proposition 1. *Let (x^k) be a sequence generated by Algorithm* (2)*. Then the sequence $(F(x^k))$ monotonically decreases and converges.*

Proof. Let x^{k+1} be a local minimum of $F^k(x)$. According to the Karush-Kuhn-Tucker (KKT) condition, there exist Lagrange multipliers $q^{k+1} \in \mathcal{H}_1$, such that

$$0 \in \partial_x \mathcal{L}_{F^k}(x^{k+1}, q^{k+1}),$$

where $\mathcal{L}_{F^k}(x, q) := F^k(x) - \langle q, Ax - b\rangle$ is the Lagrangian function. Equivalently,

$$A^\top q^{k+1} \in \partial F^k(x^{k+1}) = \partial F_1(x^{k+1}) + w^k \cdot \partial |x^{k+1}|.$$

This means that there exist vectors $d^{k+1} \in \partial F_1(x^{k+1})$, $c^{k+1} \in \partial |x^{k+1}|$ such that

$$d^{k+1} = A^\top q^{k+1} - w^k \cdot c^{k+1}. \tag{4}$$

[1]The *superdifferential* $\overline{\partial}$ of a concave function F is an equivalent of subdifferential of convex functions and can be defined by $\overline{\partial} F = -\partial(-F)$, since $-F$ is convex.

We use this to rewrite the function difference as follows:

$$\begin{aligned} &F(x^k) - F(x^{k+1}) \\ &= F_1(x^k) - F_1(x^{k+1}) + F_2(|x^k|) - F_2(|x^{k+1}|) \\ &\geq (d^{k+1})^\top (x^k - x^{k+1}) + (w^k)^\top (|x^k| - |x^{k+1}|) \\ &= (A^\top q^{k+1})^\top (x^k - x^{k+1}) \\ &\quad + (w^k)^\top (|x^k| - |x^{k+1}| - c^{k+1} \cdot (x^k - x^{k+1})) \\ &= (q^{k+1})^\top (Ax^k - Ax^{k+1}) + (w^k)^\top (|x^k| - c^{k+1} \cdot x^k) \\ &= (q^{k+1})^\top (b - b) + \sum_i w_i^k (|x_i^k| - c_i^{k+1} x_i^k) \geq 0, \end{aligned} \tag{5}$$

which means that the sequence decreases. Here in the first inequality we use the definitions of sub- and superdifferential, in the following transition we use (4). In the next-to-last transition we use that x^k and x^{k+1} are both solutions of the constrained problem (2) and $c^{k+1} \cdot x^{k+1} = |x^{k+1}|$ by definition of c^{k+1}. The last inequality follows from the fact that $w_i^k \geq 0$ and $|x_i^k| \geq c_i^{k+1} x_i^k$, as $|c_i^{k+1}| \leq 1$.

The sequence $(F(x^k))$ decreases and, by property of F, is bounded from below. Hence, it converges. □

Proposition 2. *Let (x^k) be a sequence generated by Algorithm* (2) *and suppose*

$$F(x) \to \infty, \quad \textit{whenever } \|x\| \to \infty \textit{ and } Ax = b, \tag{6}$$

then the sequence (x^k) is bounded and has at least one accumulation point.

Proof. By Proposition 1, the sequence $(F(x^k))$ is monotonically decreasing, therefore the sequence (x^k) is contained in the level set

$$L(x^0) := \{x : F(x) \leq F(x^0)\}.$$

From Property (6) of F we conclude boundedness of the set $L(x^0) \cap \{x : Ax = b\}$. This allows to apply the Theorem of Bolzano-Weierstraß, which gives the existence of a converging subsequence and, hence, an accumulation point. □

For further analysis we need F_2 to fulfill the following conditions:

(C1) F_2 is twice continuously differentiable in $\mathcal{H}_+$ and there exists a subspace $\mathcal{H}_c \subset \mathcal{H}$ such that for all $x \in \mathcal{H}_+$ holds: $h^\top \partial^2 F_2(x) h < 0$ if $h \in \mathcal{H}_c$ and $h^\top \partial^2 F_2(x) h = 0$ if $h \in \mathcal{H}_c^\perp$.

(C2) $F_2(|x|)$ is a C^1-perturbation of a convex function, i.e. can be represented as a sum of a convex function and a C^1-smooth function.

Lemma 1. *Let (x^k) be a sequence generated by the Algorithm* (2) *and suppose (x^k) is bounded and Condition* (C1) *holds for F_2. Then*

$$\lim_{k\to\infty} (\overline{\partial} F_2(|x^k|) - \overline{\partial} F_2(|x^{k+1}|)) = 0. \tag{7}$$

Proof. See supplementary material. □

Proposition 3. *Let (x^k) be a sequence generated by Algorithm* (2) *and Condition* (6) *be satisfied.*

Suppose x^* is an accumulation point of (x^k). If the function F_2 fulfills Conditions (C1) *and* (C2)*, then x^* is a stationary point*[2] *of* (1)*.*

Proof. Proposition 2 states the existence of an accumulation point x^* of (x^k), i.e., the limit of a subsequence (x^{k_j}). From (4) we have:

$$0 = d^{k_j} + \overline{\partial} F_2(|x^{k_j-1}|) \cdot c^{k_j} - A^\top q^{k_j}.$$

Combining this with (7) of Lemma 1 we conclude

$$\lim_{j\to\infty} \xi^j = 0, \quad \xi^j := d^{k_j} + \overline{\partial} F_2(|x^{k_j}|) \cdot c^{k_j} - A^\top q^{k_j}.$$

It's easy to see that $\xi^j \in \partial \mathcal{L}_F(x^{k_j})$. By Condition (C2) and a property of subdifferential of a C^1-perturbation of a convex function [11, Remark 2.2] we conclude that $0 \in \partial_x \mathcal{L}_F(x^*)$. From $Ax^{k_j} = b$ it immediately follows that $Ax^* = b$, i.e., $0 \in \partial_q \mathcal{L}_F(x^*)$, which concludes the proof. □

4. Computer vision applications

For computer vision tasks we formulate a specific subclass of the generic problem (1) as:

$$\begin{aligned} \min_{Ax=b} F(x) &= \min_{Ax=b} F_1(x) + F_2(|x|) \\ &:= \min_{Ax=b} \|Tx - g\|_q^q + \Lambda^\top \mathbf{F}_2(|x|), \end{aligned} \tag{8}$$

where $\mathbf{F}_2 \colon \mathcal{H}_+ \to \mathcal{H}_+$ is a coordinate-wise acting increasing and concave function, $A \colon \mathcal{H} \to \mathcal{H}_1$, $T \colon \mathcal{H} \to \mathcal{H}_2$ are linear operators acting between finite dimensional Hilbert spaces $\mathcal{H}$ and $\mathcal{H}_1$ or $\mathcal{H}_2$. The weight $\Lambda \in \mathcal{H}_+$ has non-negative entries. The *data-term* is the convex ℓ_q-norm with $q \geq 1$. Prototypes for $\mathbf{F}_2(|x|)$ are

$$|x_i| \mapsto (|x_i| + \varepsilon)^p \quad \text{or} \quad |x_i| \mapsto \log(1 + \beta|x_i|), \quad \forall i, \tag{9}$$

i.e., the *regularized ℓ_p-norm*, $0 < p < 1$, $\varepsilon \in \mathbb{R}_+$, or a non-convex log-*function* (c.f. Figure 2). In the sequel, the inner product $F_2 = \Lambda^\top \mathbf{F}_2$ uses either of these two coordinate-wise strictly increasing *regularization-terms*. The ℓ_p-norm becomes Lipschitz by the ε-regularization and the log-function naturally is Lipschitz.

Algorithm (2) simplifies to

$$x^{k+1} = \arg\min_{Ax=b} \|Tx - g\|_q^q + \|\mathrm{diag}(\Lambda)(w^k \cdot x)\|_1, \tag{10}$$

where the weights given by the superdifferential of F_2 are

$$w_i^k = \frac{p}{(|x_i^k| + \varepsilon)^{1-p}} \quad \text{or} \quad w_i^k = \frac{\beta}{1 + \beta|x_i|}, \tag{11}$$

[2]Here by stationary point we mean x^* such that $0 \in \partial \mathcal{L}_F(x^*)$.

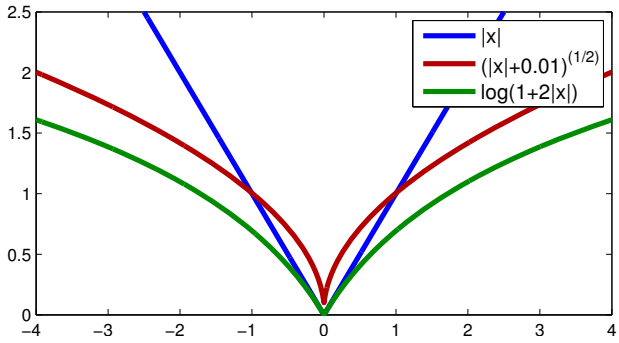

Figure 2. **Top right:** Non-convex functions of type (9): ℓ_1-norm, ℓ_p-norm with $p = 1/2$, and log-function with $\beta = 2$.

respectively. By construction, Proposition 1 applies and $(F(x^k))$ is monotonically decreasing. Proposition 2 guarantees existence of an accumulation point given Condition (6) being true. This is crucial for solving the optimization problem. The following Lemma reduces Condition (6) to a simple statement about the intersection of the kernels ker of the operators T and $\mathrm{diag}(\Lambda)$ with the affine constraint space.

Lemma 2. *Let*

$$\ker T \cap \ker \mathrm{diag}(\Lambda) \cap \ker A = \{0\}. \tag{12}$$

then $F(x) \to \infty$, whenever $\|x\| \to \infty$ and $Ax = b$.

Proof. By Condition (12) we have

$$\begin{aligned} \ker A = (\ker T \cap \ker A) \oplus (\ker \mathrm{diag}(\Lambda) \cap \ker A) \\ \oplus (\ker A / ((\ker T \oplus \ker \mathrm{diag}(\Lambda)) \cap \ker A)). \end{aligned} \tag{13}$$

For any x such that $Ax = b$ this gives $x = x^0 + e^1 + e^2 + e^3$, where x^0 is a fixed point such that $Ax^0 = b$ and e^i lie in respective subspaces from the decomposition (13). If $\|x\| \to \infty$, then $\max_i \|e^i\| \to \infty$. It is easy to see that then the maximum of summands in (8) goes to infinity. □

Considering Proposition 3; as our prototypes (9) are one-dimensional it is easy to see that (C1) and (C2) are satisfied (c.f. Lemma 3 of supplementary material for details). Therefore, only Condition (12) needs to be confirmed in order to make full use of the results proved in Subsection 3.2.

In the sequel, for notational convenience, let I_u be the identity matrix of dimension $\dim(u) \times \dim(u)$. The same applies for other operators, e.g., T_u be an operator of dimensions such that it can be applied to u, i.e., a matrix with range in a space of dimension $\dim(u)$.

Using this convention, we set in (8)

$$x = (u, v)^\top, \; T = \begin{pmatrix} T_u & 0 \\ 0 & 0 \end{pmatrix}, \; g = (g_u, 0)^\top,$$

$$\Lambda = (0, (1/\lambda)_v)^\top, \; A = \begin{pmatrix} K_u & -I_v \end{pmatrix}, \; b = (0, 0)^\top,$$

where T is a block matrix with operator T_u and zero blocks. This yields a template for typical computer vision problems:

$$\min_u \lambda \|T_u u - g_u\|_q^q + F_2(|K_u u|). \tag{14}$$

The Criterion (12) in Lemma 2 simplifies.

Corollary 1. *Let T_u be injective on $\ker K_u$. Then, the sequence generated by* (10) *is bounded and has at least one accumulation point.*

Proof. The intersection in Condition (12) equals

$$\begin{aligned}&\ker T \cap \ker \operatorname{diag}(\Lambda) \cap \ker A\\ &= \{(u,v)^\top : u \in \ker T_u\} \cap \{(u,0)^\top\}\\ &\quad \cap \{(u,v)^\top : K_u u = v\}\\ &= \{(u,0)^\top : u \in \ker T_u \wedge u \in \ker K_u\},\end{aligned}$$

where the latter condition is equivalent to T_u being injective on $\ker K_u$. Lemma 2 and Proposition 2 apply. □

Examples for the operator K_u are the gradient or the learned prior [15]. For $K_u = \nabla_u$[3] the condition from the Corollary is equivalent to $T_u \mathbf{1}_u \neq 0$, where $\mathbf{1}_u$ is the constant 1-vector of same dimension as u.

We also explore a non-convex variant of TGV [6]

$$\min_{u,w} \lambda \|T_u u - g_u\|_q^q + \alpha_1 F_2(|\nabla_u u - w|) + \alpha_2 F_2(|\nabla_w w|), \tag{15}$$

or as constrained optimization problem

$$\begin{aligned}\min_{u,w,z_1,z_2} \; & \|T_u u - g_u\|_q^q + \frac{\alpha_1}{\lambda} F_2(|z_1|) + \frac{\alpha_2}{\lambda} F_2(|z_2|)\\ \text{s.t. } & z_1 = \nabla_u u - w\\ & z_2 = \nabla_w w,\end{aligned} \tag{16}$$

which fits to (8) by setting

$$x = (u, w, z_1, z_2)^\top,\ T = \begin{pmatrix} T_u & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix},$$

$$g = (g_u, 0, 0, 0)^\top,\ \Lambda = (0, 0, (\alpha_1/\lambda)_{z_1}, (\alpha_2/\lambda)_{z_2})^\top,$$

$$A = \begin{pmatrix} \nabla_u & -I_w & I_{z_1} & 0\\ 0 & \nabla_w & 0 & -I_{z_2}\end{pmatrix},\ b = (0,0)^\top.$$

The corresponding statement to Corollary 1 is:

Corollary 2. *If T_u is injective on $\{u : \exists t \in \mathbb{R}\colon \nabla_u u = t\mathbf{1}_u\}$, then the sequence generated by* (10) *is bounded and has at least one accumulation point.*

[3] ∇_u does not mean the differentiation with respect to u, but the gradient operator such that it applies to u, i.e, ∇_u has dimension $2\dim(u) \times \dim(u)$ for 2D images.

Proof. The intersection in Condition (12) equals

$$\begin{aligned}&\{(u,w,z_1,z_2)^\top : u \in \ker T_u\} \cap \{(u,w,0,0)^\top\}\\ &\quad \cap \{(u,w,z_1,z_2)^\top : z_1 = \nabla_u u - w \wedge z_2 = \nabla_w w\}\\ &= \{(u,w,0,0)^\top : u \in \ker T_u \wedge \nabla_u u = w \wedge \nabla_w w = 0\},\end{aligned}$$

which implies the statement by Lemma 2 and Proposition 2. □

4.1. Algorithmic realization

As the inner problem (10) is a convex minimization problem, it can be solved efficiently, e.g., [18, 3]. We use the algorithm in [8, 21]. It can be applied to a class of problems comprising ours and has proved optimal convergence rate: $O(1/n^2)$ when F_1 or F_2 from (8) is uniformly convex and $O(1/n)$ for the more general case.

We focus on the (outer) non-convex problem. Let $(x^{k,l})$ be the sequence generated by Algorithm (2), where the index l refers to the inner iterations for solving the convex problem, and k to the outer iterations. Proposition 1, which proves $(F(x^{k,0}))$ to be monotonically decreasing, provides a natural stopping criterion for the inner and outer problem. We stop the inner iterations as soon as

$$F(x^{k,l}) < F(x^{k,0}) \quad \text{or} \quad l > m_i, \tag{17}$$

where m_i is the maximal number of inner iterations. For a fixed k, let l_k the number of iterations required to satisfy the inner stopping criterion (17). Then, outer iterations are stopped when

$$\frac{F(x^{k,0}) - F(x^{k+1,0})}{F(x^{0,0})} < \tau \quad \text{or} \quad \sum_{i=0}^{k} l_i > m_o, \tag{18}$$

where τ is a threshold defining the desired accuracy and m_o the maximal number of iterations. As default value we use $\tau = 10^{-6}$ and $m_o = 5000$. For strictly convex problems we set $m_i = 100$, else, $m_i = 400$. The difference in (18) is normalized by the initial function value to be invariant to a scaling of the energy. When we compare to ordinary convex energies we use the same τ and m_o.

The tasks in the following subsections are implemented in the unconstrained formulation. The formulation as a constrained optimization problem was used for theoretical reasons. In all figures we compare the non-convex norm with its corresponding convex counterpart. We always try to find a good weighting (usually λ) between data and regularization term. We do not change the ratio between weights among regularizers as for TGV (α_1 and α_2).

4.2. Image denoising

We consider the extension of the well-known Rudin, Osher, and Fatemi (ROF) model [24] to non-convex norms

$$\min_u \frac{\lambda}{2} \|u - g_u\|_2^2 + F_2(|K_u u|),$$

Figure 3. Natural image denoising problem. Displayed is the zoom into the right part of watercastle. Non-convex norms yield sharper discontinuities and show superiority with respect to their convex counterparts. **From left to right:** Original image, degraded image with Gaussian noise with $\sigma = 25$. Denoising with TGV prior, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\lambda = 5$ ($PSNR = 27.19$), and non-convex log TGV prior with $\beta = 2$, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\lambda = 10$ ($PSNR = 27.87$). The right pair compares the learned prior with convex norm ($PSNR = 28.46$) with the learned prior with non-convex norm $p = 1/2$ ($PSNR = 29.21$).

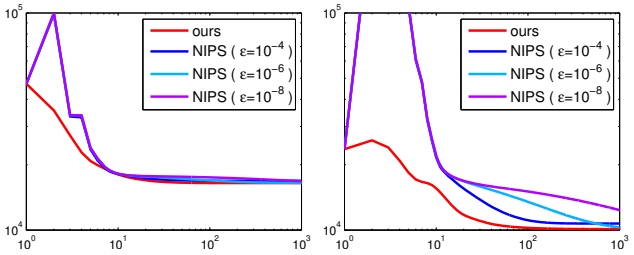

Figure 4. **Left to right:** Comparison of the energy decrease for the non-convex log TV and TGV between our method and NIPS [25]. Our proposed algorithm achieves a lower energy in the limit and drops the energy much faster in the beginning.

and arbitrary priors K_u, e.g., here, $K_u = \nabla_u$ or K_u from [15]. Since $\ker T_u = \ker I_u = \{0\}$ Condition (12) is trivially satisfied for all priors, c.f. Corollary 1 and 2. The regularizing norms $F_2(|x|) = \sum_i (\mathbf{F}_2(|x|))_i$ are the ℓ_p-norm, $0 < p < 1$, and the log-function according to (9).

Figure 3 compares TGV, the learned image prior from [15] and their non-convex counterparts. Using non-convex norms combines the ability to recover sharp edges and being smooth in between.

Figure 4 demonstrates the efficiency of our algorithm compared to a recent method, called, non-convex inexact proximal splitting (NIPS) [25], which is based on smoothing the objective. Reducing the smoothing parameter ε better approximates the original objective, but, on the other hand, increases the required number of iterations. This is expected as the ε directly effects the Lipschitz constant of the objective. We do not require such a smoothing epsilon and outperform NIPS.

4.3. Image deconvolution

Image deconvolution is a prototype of inverse problems in image processing with non-trivial kernel, i.e., the model is given by a non-trivial operator $T_u \neq I$ in (14) or (15). Usually, T_u is the convolution with a point spread function k_u, acting as a blur operator. The data-term here reads $\|k_u * u - g_u\|_q^q$. Obviously $\ker k_u = \{0\}$ and Corollaries 1 and 2 are fulfilled. We assume Gaussian noise, hence, we use $q = 2$.

We use the numerical scheme of [8] based on the fast Fourier transform to implement the data-term and combine it with the non-convex regularizers.

Deconvolution aims for the restoration of sharp discontinuities. This makes non-convex regularizers particularly attractive. Figure 5 compares different regularization terms.

4.4. Optical flow

We estimate the optical flow field $u = (u^1, u^2)^\top$ between an image pair $f(x,t)$ and $f(x,t+1)$ according to the energy functional:

$$\min_{u,v,w} \lambda \|\rho(u,w)\|_1 + \|\nabla_w w\|_1 + \alpha_1 F_2(|\nabla_u u - v|) + \alpha_2 F_2(|\nabla_v v|),$$

where local brightness changes w between images are assumed to be smooth [8]:

$$\rho(u,w) = f_t + (\nabla_f f)^\top \cdot (u - u_0) + \gamma w.$$

We define $\nabla_u u = (\nabla_{u^1} u^1, \nabla_{u^2} u^2)^\top$, and v according to the definition of TGV.

A popular regularizer is the total variation of the flow field. However, this assigns penalty to a flow field describing rotation and scaling motion. TGV regularization deals with this problem and affine motion can be described without penalty. Figure 6 shows that enforcing the TGV properties by using non-convex norms yields highly desirable sharp motion discontinuities and convex TGV regularization is outperformed.

Since we analysed TGV already for Condition (12) only

Figure 5. Deconvolution example with known blur kernel. Shown is a zoom to the right face part of romy. **From left to right:** Original image, degraded image with motion blur of length 30 rotated by 45° and Gaussian noise with $\sigma = 5$. Deconvolution using TGV with $\lambda = 400$, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$ ($PSNR = 29.92$), non-convex log-TGV, $\beta = 1$, with $\lambda = 300$, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$ ($PSNR = 30.15$), the learned prior [15] with $\lambda = 25$ ($PSNR = 29.71$), and its non-convex counterpart with $p = 1/2$ and $\lambda = 40$ ($PSNR = 30.54$).

the data-term is remaining. We obtain (8) by setting

$$T = \begin{pmatrix} \operatorname{diag}((\nabla_f f)^\top) & \gamma I_w \\ 0 & \nabla_w \end{pmatrix}, \; g = -\begin{pmatrix} f_t - (\nabla_f f)^\top \cdot u_0 \\ 0 \end{pmatrix}$$

and the kernel of T can be estimated as

$$\begin{aligned} \ker T &= \{(u,w)^\top : T(u,w)^\top = 0\} \\ &= \{(u,w)^\top : (\nabla_f f)^\top \cdot u = -\gamma w \wedge \nabla_w w = 0\} \\ &= \{(u, t\mathbf{1}_w)^\top : (\nabla_f f)^\top \cdot u = -\gamma t \mathbf{1}_w, \; t \in \mathbb{R}\}. \end{aligned}$$

For TV and TGV this requires a constant or linear dependency for x- and y-derivative of the image for all pixels. Practically interesting image pairs do not have such a fixed dependency, i.e., Lemma 2 applies.

4.5. Depth map fusion

In the non-convex generalization of TGV depth fusion from [22] the goal is minimize

$$\lambda \sum_{i=1}^{K} \|u - g_i\|_1 + \alpha_1 F_2(|\nabla_u u - v|) + \alpha_2 F_2(|\nabla_v v|)$$

with respect to u and v, where the g_i, $i = 1, \dots, K$, are depth maps recorded from the same view. The data-term in (8) is obtained by setting $T = (T_{u_1}, \dots, T_{u_K})^\top$ and $T_{u_i} = I_{u_i}$, the identity matrix. Hence, Condition (12) is satisfied.

Consider Figure 7; the streets, roof, and also the round building in the center are much smoother for the result with non-convex norm, and, at the same time discontinuities are not smoothed away, they remain sharp (c.f. Figure 1).

5. Conclusion

The iteratively reweighted ℓ_1 minimization algorithm for non-convex sparsity related problems has been extended to a much broader class of problems comprising computer vision tasks like image denoising, deconvolution, optical flow estimation, and depth map fusion. In all cases we could show favorable effects when using non-convex norms.

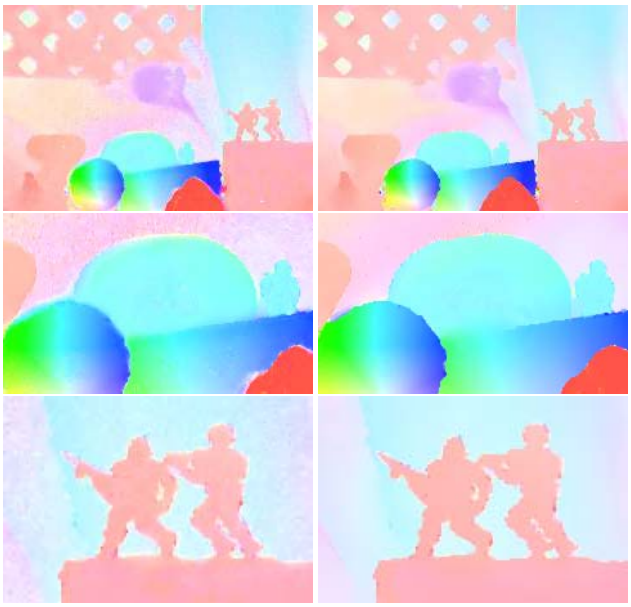

Figure 6. Comparison between TGV (**left**) and non-convex TGV (**right**) for the image pair *Army* from the Middlebury Optical Flow benchmark [1] and two zooms. The TGV is obtained with $\lambda = 50$, $\gamma = 0.04$, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$ and the result with non-convex TGV using $p = 1/2$, $\lambda = 40$, $\gamma = 0.04$, $\alpha_1 = 0.5$, $\alpha_2 = 1.0$.

The presentation of an efficient optimization framework for the considered class of linearly constrained non-convex non-smooth optimization problems has been enabled by proving decreasing function values in each iteration, the existence of accumulation points, and boundedness.

Acknowledgment

This project was funded by the German Research Foundation (DFG grant BR 3815/5-1), the ERC Starting Grant VideoLearn, and by the Austrian Science Fund (project P22492).

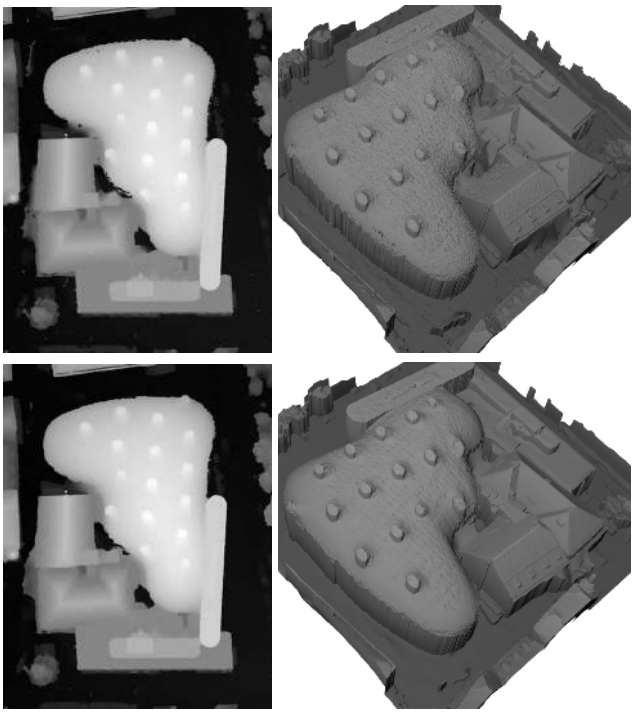

Figure 7. Non-convex TGV regularization (**bottom row**) yields a better trade-off between sharp discontinuities and smoothness than its convex counterpart (**upper row**) for depth map fusion. **Left:** Depth maps. **Right:** Corresponding rendering.

References

[1] Middlebury optical flow benchmark. `vision.middlebury.edu`.

[2] L. An and P. Tao. The DC (Difference of Convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133:23–46, 2005.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Applied Mathematics*, 2(1):183–202, 2009.

[4] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.

[5] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[6] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Applied Mathematics*, 3(3):492–526, 2010.

[7] E. J. Candes, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier Analysis and Applications,*, 2008.

[8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[9] X. Chen and W. Zhou. Convergence of the reweighted ℓ_1 minimization algorithm. Preprint, 2011. Revised version.

[10] I. Daubechies, R. Devore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[11] M. Fornasier and F. Solombrino. Linearly contrained nonsmooth and nonconvex minimization. Preprint, 2012.

[12] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:367–383, 1992.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[14] J. Huang and D. Mumford. Statistics of natural images and models. In *International Conference on Computer Vision and Pattern Recognition*, pages 541–547, USA, 1999.

[15] K. Kunisch and T. Pock. A bilevel optimization approach for parameter learning in variational models. *Preprint*, 2012.

[16] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[17] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[18] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[19] M. Nikolova and R.H.Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Transactions on Image Processing*, 16(6):1623–1627, 2007.

[20] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *Proc. IEEE Computer Society Workshop on Computer Vision*, pages 16–22, Miami Beach, FL, 1987. IEEE Computer Society Press.

[21] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *International Conference on Computer Vision*, 2011.

[22] T. Pock, L. Zebedin, and H. Bischof. Tgv-fusion. *C.S. Calude, G. Rozenberg, A. Salomaa (Eds.): Maurer Festschrift, LNCS 6570*, pages 245–258, 2011.

[23] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[24] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[25] S. Sra. Scalable nonconvex inexact proximal splitting. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 539–547. 2012.

[26] C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Transactions on Image Processing*, 7:813–824, 1998.