# On iteratively reweighted Algorithms 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

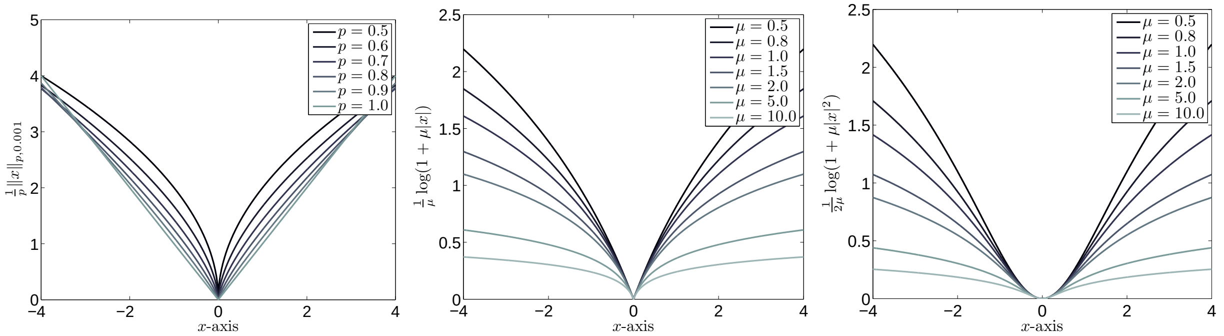
[2] Graz University of Technology,
Austria
pock@icg.tugraz.at

January 10, 2015

**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 of optimization. Recently, iteratively reweighed $\ell_1$ minimization (IRL1) has been proposed as a way to tackle a class of non-convex functions by solving a sequence of convex $\ell_2$-$\ell_1$ problems. We extend the problem class to the sum of a convex function and a (non-convex) non-decreasing function applied to another convex function. The proposed algorithm sequentially optimizes suitably constructed convex majorizers. Convergence to a critical point is proved when the Kurdyka-Łojasiewicz property and additional mild restrictions hold for the objective function. The efficiency of the algorithm and the practical importance of the algorithm is demonstrated in computer vision tasks such as image denoising and optical flow. Most applications seek smooth results with sharp discontinuities. This is achieved by combining non-convexity with higher order regularization.

***Keywords*** — *Iteratively reweighted algorithm, majorization-minimization, IRL1, IRLS, non-smooth non-convex optimization, Kurdyka-Łojasiewicz inequality, computer vision, non-convex total generalized variation*



(a) $\frac{1}{p}\|x\|_p$

(b) $\frac{1}{\mu}\log(1+\mu|x|)$

(c) $\frac{1}{\mu}\log(1+\mu x^2)$

Figure 1: Non-convex prototype functions for $F_2$.

# 1 Introduction

In the last decade we have seen a strong interest in the development of efficient first-order algorithms for minimizing structured convex optimization problems [61, 8, 28, 23, 62]. Nowadays, we have algorithms at hand that can be applied to efficiently solve convex optimization problems frequently arising in computer vision, signal processing and machine learning problems.

However, while most of these problems can be modeled with sufficient accuracy using convex objective functions, it is also well-known that certain desirable modeling aspects such as the robustness to noise, the recovery of sparse and discontinuous signals require the use of non-convex objective functions. The goal of this paper is therefore to develop efficient optimization algorithms that can be applied to a certain class of non-smooth and non-convex objective functions of the form:

$$\min_{x\in X} F_1(x) + F_2(G(x))\,, \tag{1}$$

where $F_1$ is a convex function, $G$ is a coordinate-wise convex function and $F_2$ is non-convex. Figure 1 shows non-convex functions $F_2$ that are particularly interesting in applications. The structure of (1) differs from related convex problems, for which efficient algorithms are available, only in $F_2 \circ G$ possibly being non-convex. One would expect that such a strong analogy can be exploited. Indeed, for the algorithm we propose in this paper we can show some favorable properties, including convergence of the function values and under some more assumptions convergence of the sequence of arguments. The numerical analysis demonstrates efficiency and robustness towards local optima. At the same time, the algorithm allows us to deal with several interesting non-convex problems in image processing.

The proposed algorithm is in the fashion of classical majorization-minimization algorithms. It generates and solves a sequence of convex optimization problems. The non-convex part $F_2$ is at each iteration approximated by means of a majorizing convex surrogate function. Then, the resulting convex optimization problem is solved. As a matter of fact, the convex surrogate function has a structure that is amenable to efficient first-order methods for structured convex optimization.

Although the convex subproblems are known to converge, it is not trivial to prove the convergence for the overall non-convex problem. We show two convergence results. The first one establishes convergence for a subsequence of the sequence generated by our algorithm. This result is easy to obtain and mainly stems from the fact that majorization-minimization algorithms generate a sequence of non-increasing function values. The second result states the convergence of the *whole* sequence. It requires a more sophisticated analysis and some more assumptions like Lipschitz continuity of the gradient of $F_2$. For a subclass of Problem (1) satisfying these assumptions, convergence of the whole sequence of arguments to a critical point is proved. One part of the stronger regularity assumption is that the objective is a KL-function, i.e., the Kurdyka-Łojasiewicz inequality holds on the whole domain. This implies a sufficient descent property for gradient based methods also in areas where the function is flat around local optima. Our approach to proof convergence is showing that the requirements of [6] are satisfied.

In our numerical experiments we will show that the performance of the proposed algorithms is comparable to related algorithms for convex optimization, e.g., gradient descent or forward-backward splitting algorithms. Moreover, the proposed algorithms are easy to implement, which make them interesting for practical applications.

Problems like (1) arise frequently in image processing, computer vision or machine learning. The applicability of the iteratively reweighted $\ell_1$ algorithm, which arises as a special case of the algorithm presented in this paper, we already demonstrated in an earlier conference paper [68]. Whereas the focus in the conference paper was the diversity of applications such as denoising, deconvolution, depth map fusion, and optical flow, in this paper we concentrate on the difference of modeling concepts in denoising and optical flow estimation. However, the concepts how the non-convex penalty functions are used easily generalize to many other problems, e.g., deconvolution, depth map fusion, stereo estimation, or superresolution. Replacing convex penalty functions by non-convex functions usually leads to better results. In particular, we analyze robust data-terms and the usage of edge-enhancing non-convex penalizers. As a special instance, we are the first to propose a non-convex extension of the total generalized variation regularizer [17]. The total generalized variation (TGV) semi-norm is a convex penalizer that can reconstruct piecewise smooth functions. Due to the convexity of the regularizer, first- and higher-order discontinuities are only preserved but not enhanced. This may lead to over-smoothing effects in case of strong noise or weak data terms. It turns out that this effect can be partly avoided by using non-convex penalizers in the TGV semi-norm.

## 2 Related work

Since the seminal works of Geman and Geman [42], Blake and Zissermann [12], and Mumford and Shah [59] on image restoration, the application of non-convex potential functions in variational approaches for computer vision problems has become a standard paradigm. Non-convexity can be motivated and justified from different viewpoints, including robust statistics [11], nonlinear partial differential equations [69], and natural image statistics [48]. Since then, numerous works demonstrated empirically [11, 77], that non-convex potential functions are the right choice for most regularization tasks in computer vision.

However, the downside is optimization. While there has been vast progress in convex optimization - today, many non-smooth convex optimization programs can be solved with comparable efficiency to linear programs - non-convex optimization is still rarely applied in practice. 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”.

Gradient descent based methods Steepest Descent, Quasi-Newton or Newton methods [60, 65, 10] are the classical approaches for general optimization and are also applicable in the non-convex setting as long as the objective is smooth enough. An alternative are hill-climbing methods [79], annealing-type schemes [42], or graduated non-convexity (GNC) [12, 63]. However, efficiency of these methods leaves room for improvement. The worst-case complexity bound for general non-convex problems derived in [60] supports this statement. This means that there is only hope for efficient algorithms when considering non-convex optimization problems of a specific structure.

In convex optimization many efficient algorithms like Douglas-Rachford [33, 34], forward-backward splitting [55, 29, 8, 60], primal-dual approaches [23, 70, 45], or augmented Lagrangian method [10, 46, 74] originate from the proximal point algorithm [58, 75].

While it seems to be difficult to generalize primal-dual approaches to the non-convex setting directly, the augmented Lagrangian method is considered in [38, 3], the gradient projection method in [54, 44, 6], or a forward-backward splitting in [39, 78, 62, 6] were used for non-convex optimization. In [67] the iPiano algorithm is introduced in the non-convex setting. It combines ideas from the forward-backward splitting and the Heavy-ball method from Polyak [73]. In our numerical experiments we will consider this algorithm, because several algorithms like gradient descent, projected gradient descent, the Heavy-ball method, and the forward-backward algorithm are special cases of it.

From another perspective, gradient descent method can also be interpreted as a majorization minimization (MM) algorithm [9, 53, 49]. The iteration step of the gradient descent method is equivalent to minimizing an isotropic quadratic upper bound—the quadratic upper bound that appears in the Descent Lemma (see e.g. Lemma 2). This way many algorithms can be considered as special instances of the MM algorithm. Also expectation-maximization algorithms are MM algorithms [35, 36, 51]. The algorithm generates a sequence of simpler majorizing functions and computes the next iterate as a minimizer of this surrogate function. The MM principle can also be used for analytically estimating step size parameters for a given search direction (on a subspace) [25, 26, 37]. In this context majorizers are mostly quadratic functions.

An important sub-class of the MM algorithms related to our algorithm is by Geman and Reynolds [41]. 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. [81, 64, 31]. Convergence results can be found in [50, 1].

The IRLS algorithm can only be applied if the non-convex function can be well approximated from above with quadratic functions. However, this does not cover interesting functions such as $\log(1+|x|)$ that are non-differentiable at zero. Candes et al. [21] tackled this problem by the so-called iteratively reweighted $\ell_1$ (IRL1) algorithm. It solves a sequence of non-smooth $\ell_1$ 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 $\ell_1$ regularized compressed sensing problems.

First convergence results for the IRL1 algorithm have been obtained by Chen et al. in [24] for a class of non-convex $\ell_2$-$\ell_p$ 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.

In our previous work [68], the convergence analysis of [24] was generalized to linearly constrained optimization problems. This analysis made the algorithm and the theoretical results applicable to many computer vision problems. In the present paper, the algorithm will be generalized further and the convergence analysis will be extended a lot compared to [24, 68]. The convergence result is based on the analysis of an abstract descent algorithm [6] and requires the objective function to be a Kurdyka-Łojasiewicz (KL) function, i.e. to satisfy the KL inequality (see Appendix). See [56, 57, 52] for smooth KL-functions and [14, 15] for non-smooth functions. Almost all functions used in computer vision are KL functions. For the examples considered in this paper the property will explicitly be verified. For information about the abstract classification of KL functions, we refer to [6, 16, 52] and references therein.

During the last years, using the Kurdyka-Łojasiewicz property, several algorithms have been shown to converge [27, 6, 4, 5]. Also the iPiano algorithm considered in the comparison in this paper was recently shown to converge if the objective function has the KL property [67]. As a last class of problems, where the KL property allowed to show convergence is the class of DC-programming [80, 2, 47] (non-convex functions that can be written as the difference of convex functions). The algorithm proceeds by constructing a sequence of convex optimization programs by linearizing sequentially one of the two convex functions.

Another line of research for solving non-convex optimization problems, where algorithm's convergence is not in doubt, is by minimizing the convex envelope of the objective [30, 22, 71, 43]. Here, the challenge is the construction of a suitable convex envelope and efficient minimization due to the (usually) increasing number of dimensions.

# 3 The model

We study a non-convex optimization problem of a specific structure in a finite dimensional real vector space $X$ of dimension $\dim(X) = n \in \mathbb{N}$. The standard inner product and norm are denoted $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_2^2 := \langle \cdot, \cdot \rangle$, respectively. The optimization problem reads

$$\min_{x\in X} F(x) := \min_{x\in X} F_1(x) + F_2(G(x))\,, \tag{2}$$

with a lower semi-continuous (lsc), extended real-valued, proper function $F\colon X \to \overline{\mathbb{R}}$ where $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$. In addition we assume that $F$ is bounded from below, i.e., $\inf_{x\in X} F(x) =: \underline{F} > -\infty$. We require that $F_1\colon X \to \overline{\mathbb{R}}$ is proper, lsc, convex. Note that we explicitly allow the function $F_1$ to take on values at infinity, hence it can be for example the indicator function of a convex set. The function $G\colon X \to X_2$ maps from $X$ into another finite dimensional real vector space $X_2$ with dimension $n_2 := \dim(X_2) \le n$. We assume each coordinate function $G_i$, $i = 1, \ldots, n_2$, to be convex. The function $F_2\colon G(X) \to \mathbb{R}$ we assume coordinate-wise non-decreasing, i.e., $F_2(x) \le F_2(x + \lambda e_i)$ whenever $x, x + \lambda e_i \in G(X)$ and $\lambda > 0$, where $e_i$ is the $i$-th standard basis vector of $X_2$, $i = 1, \ldots, n_2$. We note that coordinate-wise convexity of $G$ gives very simple structure to the set $G(X)$: it is Cartesian product of intervals, each infinite on one or both ends.

**Example 1** (Denoising)**.** Image denoising is a simple example from image processing that fits into the framework of (2). Given a noisy image $f \in X$ the goal of denoising is to find the image $u \in X$ such that $f = u + h$, where $h \in X$ is the noise that deteriorated the recording. Commonly, $u$ instead of $x$ denotes the optimization variable in image processing. The denoised image, i.e. the result, can be sought as minimizer of

$$\min_{u\in X} \lambda\|u - f\|_1 + \sum_i \log(1 + |Du|_i)\,,$$

where $\lambda \in \mathbb{R}_+$ and $|Du| \in G(X)$ denotes the vector of coordinates $|Du|_i := \sqrt{((D_x u)_i)^2 + ((D_y u)_i)^2}$, where $D_x u$ is a discrete implementation of the $x$-derivative of the image (considered as a function $\mathbb{R}^2 \to \mathbb{R}$). The first term measures the discrepancy between the measurements and the sought denoised image. It is called *data-term* and given by the proper, convex (non-smooth) function $F_1(u) = \lambda\|u - f\|_1$ in (2). The second term, called the *regularization-term* invokes some prior knowledge about natural image statistics. The use of the non-convex function $F_2(y) = \sum_i \log(1 + y_i)$ on $G(X)$ for the regularization-term stresses the general property of images of being smooth and having some sharp jump discontinuities. Obviously, $F_2$ is coordinate-wise non-decreasing on $G(X)$ (see Figure 1). The coordinate functions $G_i(u) = |Du|_i$ are convex and make $F_2 \circ G$ non-smooth.

Finding global minimum of a non-smooth non-convex function, as in (2), is in general not feasible. We hence only aim to find *a critical point* of the function $F$, i.e. $x \in X : 0 \in \partial F(x)$. Here we make use of limiting-subgradient, see Definition 4 in the Appendix. Critical points are connected to local minima of the function by the following *Fermat's rule* ([76, Thm. 10.1]):

**Theorem 1** (Fermat's rule)**.** *If a proper function $F\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ has a local minimum at $\bar{x}$, then $0 \in \partial F(x)$.*

# 4 Iterative convex majorization minimization

In this paper, we study a sub-class of majorization-minimization (MM) methods that is suitable for solving the minimization problem (2). The idea of MM algorithms is to minimize majorizers of the function instead of the function itself. The major challenge is the construction of majorizing functions that are easier to minimize than the original function. Invoking only some weak assumptions about the structure of the optimization problem (2) as done above, makes possible designing such majorizing functions.

We propose to majorize $F_2$ with a convex function $F_2^{x^k}$ that approximates $F_2$ such that $F_2^{x^k} \circ G$ is convex and meets $F_2 \circ G$ at $x^k$. More formally, consider the generic Method 1.

**Method 1** (Iterative convex majorization minimization method)**.**

- *Initialization: Choose a starting point $x^0 \in X$ with $F(x^0) < \infty$ and define a suitable family of convex surrogate functions $(F_2^x)_{x\in X}$, such that for all $x \in X$ holds $F_2^x \in \mathcal{F}_{2,G}(x)$, where*

$$\mathcal{F}_{2,G}(x) := \left\{ f\colon X_2 \to \mathbb{R} \,\middle|\, \begin{array}{c} f \text{ proper, convex,} \\ f \text{ non-decreasing on } G(X), \\ f(G(x)) = F_2(G(x)), \\ \forall y \in G(X)\colon f(y) \geq F_2(y) \end{array} \right\} . \tag{3}$$

- *Iterations $(k \geq 0)$: Update*

$$x^{k+1} = \arg\min_{x\in X} F_1(x) + F_2^{x^k}(G(x)) \,. \tag{4}$$

Non-decreasingness of $F_2^x$ provides convexity of the composition $F_2^x \circ G$. As the above formulation is rather abstract, we exemplify the algorithm.

**Example 2.** We consider a simplified problem of Example 1 and restrict $X = X_2 = \mathbb{R}$, $G(x) = |x|$:

$$\min_{x\in\mathbb{R}} \; F(x) = \min_{x\in\mathbb{R}} \; F_1(x) + F_2(x) = \min_{x\in\mathbb{R}} \; 2|x-1| + \frac{1}{2}\log(1 + 25x^2) \,. \tag{5}$$

Figure 2 visualizes one update step using (4) at $x^k = -0.5$. For details on how to choose the surrogate function we refer to the specialized algorithms (here Algorithm 5) introduced in the following subsections.

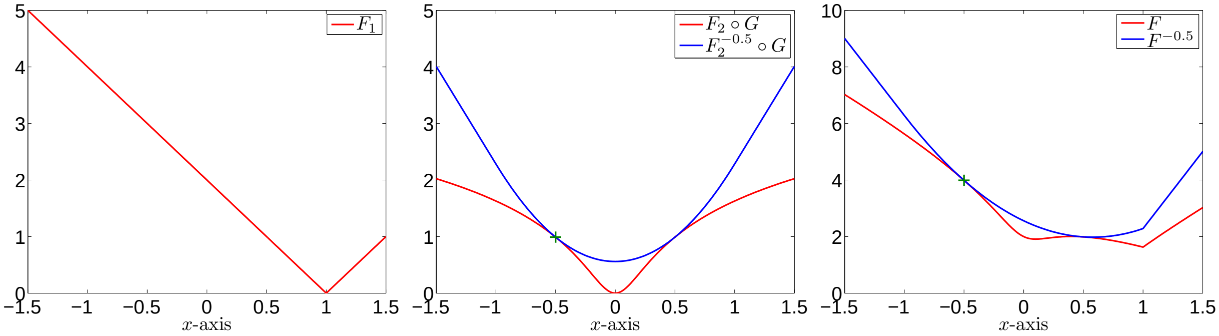


Figure 2: Visualization of one update step (4) for optimization problem (5) at $x^k = -0.5$. Left: convex function $F_1$; middle: non-convex function $F_2 \circ G$ (red) and its convex majorizer $F_2^{-0.5} \circ G$ (blue); right: the function $F = F_1 + F_2 \circ G$ (red) and its majorizer $F^{-0.5} = F_1 + F_2^{-0.5} \circ G$ (blue). The blue function in the right plot is to be minimized to obtain $x^{k+1}$.

We call Method 1 "generic", because it still requires the choice of suitable convex surrogate functions. As $F_1$ is already convex, the approximation $F_2^{x^k}$ of $F_2$ is the focus of attention. A choice of approximation follows from the following property of MM algorithms.

**Proposition 1.** *Let $(x^k)_{k\in\mathbb{N}}$ be generated by Method 1 and let for all $x \in X$ be $F_2^x \in \mathcal{F}_{2,G}(x)$. Then, the sequence $(F(x^k))_{k\in\mathbb{N}}$ monotonically decreases and converges.*

*Proof.* The proof directly follows from $F$ being bounded from below by $\underline{F}$ and the definitions of $x^k$ and $F_2^{x^k}$:

$$\underline{F} \leq F(x^{k+1}) \leq F_1(x^{k+1}) + F_2^{x^k}(G(x^{k+1})) \leq F_1(x^k) + F_2^{x^k}(G(x^k)) = F(x^k) \,.$$

The sequence $(F(x^k))_{k\in\mathbb{N}}$ decreases and is bounded from below. Hence, it converges. □

Clearly, at each iteration the value of the function $F$ decreases at least as much as the value of the majorizing function. This suggests to use surrogate functions whose minimum is minimal. Of course, it could happen that another surrogate function with a higher minimum yields a lower value of the original function, however, there is no guarantee. Finding the optimal approximation according to the *criterion of guaranteed maximal decrease of function values* is hard. In general, a majorizer $f \in \mathcal{F}_{2,G}(x^k)$ that is not the sum of $F_1$ and another convex function can have a lower minimum than our approximation. However, this better majorizer may be complex to construct and difficult to optimize. Thus, we aim to fulfill the criterion of guaranteed maximal decrease of function values in the class of surrogate functions that are sum of $F_1$ and another convex function. If we talk about *optimal majorizers* in the following, we mean optimality according to the guaranteed decrease of function values in this class. These are majorizers (constructed under certain conditions) with the lowest minimum.

The different algorithms that will be presented in the following section take into account the characteristics of $F_2$. Considering again Figure 1, it is obvious that there is no simple universal choice for the surrogate function. For instance, close to 0 the functions in Figure 1(a) and (b) may be well approximated by the absolute value function, whereas this is a bad choice for Figure 1(c). Different functions require different construction principles of the majorizer. In this paper, we show constructions of majorizers, which address (2). Some of them are proved to be optimal in the sense explained above. Method 1 serves as a tool to prove their convergence in a unified framework. However, it covers many other possible constructions (thus also convergence) of majorizers that are not explicitly presented here.

# 5 Iteratively reweighted convex algorithms

As the function $F_2$ in the optimization problem (2) does not change, it may be possible to find a single convex function that, weighted appropriately, can serve as majorizer for $F_2$ at each step of the Method 1. This is the principle of iteratively reweighted algorithms. The construction of majorizers according to this principle is easier than for the very general Method 1 and allows for explicit algorithms. The reweighting algorithms considered in the subsequent subsections are all special cases of the IRconvex Method 2, which is an instance of Method 1.

**Method 2** (Iteratively reweighted convex method: IRconvex)**.**

- *Initialization: Define a convex function* $F_2^c\colon G(X) \to \mathbb{R}^{n_2'}$, $n_2' \in \mathbb{N}$, *and a family of vectors* $(w^x)_{x\in X}$ *such that*
$$y \mapsto \langle w^x, F_2^c(y)\rangle \in \mathcal{F}_{2,G}(x), \quad x \in X\,,$$
*and starting point* $x^0 \in X$ *with* $F(x^0) < \infty$.
- *Iterations* $(k \geq 0)$*: Update*
$$x^{k+1} = \arg\min_{x\in X} F_1(x) + \left\langle w^{x^k}, F_2^c(G(x))\right\rangle \tag{6}$$

**Remark 1.** As the optimization problem in (6) is independent of constants, $y \mapsto \langle w^x, F_2^c(y)\rangle$ may be in $\mathcal{F}_{2,G}(x)$ only after adding a constant. Formally this could be achieved by setting $\tilde{F_2}^c := (F_2^c, 1)$ and $\tilde{w}^x := (w^x, a)$, where $a \in \mathbb{R}$. Being aware of it now, subsequently, we will simply neglect the constant.

## 5.1 Iteratively reweighted $\ell_1$ algorithm

Algorithm 3, the iteratively reweighted $\ell_1$ algorithm, will be shown to be optimal for the optimization problem (2) in a certain sense, when $F_2$ is concave on $G(X)$. We denote here by $\bar{\partial} f := -\partial(-f)$ the *limiting-supergradient*, the analogue of the limiting-subgradient but for concave functions. The usage of $\bar{\partial}$ instead of $\partial$ makes the algorithm slightly more general. For $F_2$ concave it is $\partial F_2(x) \subset \bar{\partial} F_2(x)$ on the interior of $G(X)$.

**Algorithm 3** (Iteratively reweighted $\ell_1$ algorithm: IRL1)**.**

- *Assumption: $F_2$ is concave on $G(X)$.*
- *Initialization: Define a family of vectors $(w^x)_{x\in X}$ with*

$$w^x \in \bar{\partial} F_2(y), \quad y = G(x), \quad x \in X\,,$$

*and starting point $x^0 \in X$ with $F(x^0) < \infty$.*

- *Iterations ($k \geq 0$): Update*

$$x^{k+1} = \arg\min_{x\in X} F_1(x) + \left\langle w^{x^k}, G(x) \right\rangle \tag{7}$$

**Remark 2.** As mentioned in Remark 1, the functions $x \mapsto \left\langle w^{x^k}, G(x)\right\rangle$ may not be majorizers of $F_2$. However, they become majorizers when shifted by suitable constants which do not affect (7). The exact majorizer will be considered in Proposition 2.

**Example 3.** We consider the same optimization problem as in Example 1. Obviously, $F_2(y) = \sum_i \log(1+y_i)$ is non-decreasing and concave on $G(X) = [0; +\infty)$, $F_1(u) = \lambda\|u - f\|_1$ and $G_i(u) = |Du|_i$ are convex. For $u^k \in X$ the vectors $w^{u^k}$ in Algorithm 3 read $w_i^{u^k} = 1/(1 + |Du^k|_i)$ which is defined as $|Du^k|_i := \sqrt{((D_x u^k)_i)^2 + ((D_y u^k)_i)^2}$, and the convex surrogate optimization problem in (7) reads

$$\min_{u\in X}\ \lambda\|u - f\|_1 + \sum_i w_i^{u^k} |Du|_i\,.$$

Each of these subproblems is a denoising problem with total variation regularization with coordinates differently weighted.

As discussed before, in general, it is hard to construct the best surrogate function according to the criterion of guaranteed maximal decrease of function values. However, assuming that $F_2$ is concave on $G(X)$ it is possible and used in Algorithm 3.

**Proposition 2.** *If $\mathcal{F}_{2,G}(x^k)$ is defined as in* (3) *and $F_2$ is concave on $G(X)$ and differentiable at $G(x^k)$, then the optimal majorizer of $F_2 \circ G$ at $x^k$*

$$\arg\min_{f\in\mathcal{F}_{2,G}(x^k)} \left(\min_{x\in X} f(G(x))\right)$$

*is given by*

$$\hat{F}_2(y) = \left\langle \nabla F_2(G(x^k)), y - G(x^k)\right\rangle + F_2(G(x^k)).$$

*Moreover, $F_1 + \hat{F}_2 \circ G$ is also the optimal majorizer of $F$ among majorizers of $F$ corresponding to majorizers of $F_2 \circ G$ from the class $\mathcal{F}_{2,G}(x^k)$.*

*Proof.* Due to concavity of $F_2$, the function $\hat{F}_2$ is a majorizer of $F_2$. It also clearly fulfills all other conditions to belong to the class $\mathcal{F}_{2,G}(x^k)$. On the other hand, for any convex function $f$ such that $f(G(x^k)) = F_2(G(x^k))$ and $f(y) \geq F_2(y)$ for all $y \in G(X)$ we have $f(y) \geq \hat{F}_2(y)$ for all $y \in G(X)$. Indeed, suppose there exists $y^*$ such that $f(y^*) < \hat{F}_2(y^*)$. Then differentiability of $F_2$ at $G(x^k)$ implies that there exists $t^* \in (0,1)$ such that

$$t^* f(y^*) + (1-t^*) f(G(x^k)) < F_2(t^* y^* + (1-t^*)G(x^k)) \leq f(t^* y^* + (1-t^*)G(x^k)).$$

This contradicts convexity of $f$, hence, our supposition was not valid, and $f(x) \geq \hat{F}_2(x)$ for all $x \in X$. Therefore, $f(G(x)) \geq \hat{F}_2(G(x))$ for all $x \in X$, which immediately gives

$$\min_{x\in X} f(G(x)) \geq \min_{x\in X} \hat{F}_2(G(x)),$$

i.e. $\hat{F}_2 \circ G$ if the best majorizer of $F_2$. Moreover, $F_1(x) + f(G(x)) \geq F_1(x) + \hat{F}_2(G(x))$ for all $x \in X$ and hence $F_1 + \hat{F}_2 \circ G$ is also the optimal majorizer of $F$. □

## 5.2 Iteratively reweighted tight convex algorithm

The iteratively reweighted $\ell_1$ Algorithm 3 is optimal for a certain class of functions, but not applicable to many other practically interesting cases, such as for example $F_2(|x|) = \log(1 + |x|^2)$ (see Figure 1). The reason is that close to 0 this prototype function is strongly convex and hence not majorized by tangents. Fortunately, the structure of this function allows for simple and tight majorizer. Namely, let us consider the *class* $\mathcal{F}_{cc}$ consisting of functions $f \colon \mathbb{R}^n_+ \to \mathbb{R}$ such that:

1. $f$ is additively separable, i.e. $f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n)$,
2. every $f_j$ is convex in the *convexity region* $[0,\, r_j]$ and concave in the *concavity region* $[r_j,\, +\infty)$ for some $r_j \geq 0$.

For simplicity we also suppose that there exist left and right derivatives $f'_j(r_j^-)$ and $f'_j(r_j^+)$. Then for $f \in \mathcal{F}_{cc}$ we denote $s_j = \max\left(f'_j(r_j^-), f'_j(r_j^+)\right)$ and define the following functions:

$$t_j(x_j) = \begin{cases} f_j(x_j), & \text{if } x_j \leq r_j, \\ f_j(r_j) + s_j(x_j - r_j), & \text{if } x_j > r_j. \end{cases} \tag{8}$$

We set $T_f(x) = (t_1(x_1), \ldots, t_n(x_n))^\top$ to be the vector of all these functions. Each $t_j$ majorizes corresponding $f_j$ because in the convexity region these two functions coincide, while in the concavity region $t_j$ majorizes the tangent $f_j(r_j) + f'_j(r_j^+)(x_j - r_j)$ of the concave function $f_j$. Moreover, each $t_j$ is convex by construction. We hence can plug $T$ into Method 2, yielding the following algorithm:

**Algorithm 4** (Iteratively reweighted tight convex algorithm: IRTight)**.**

- *Assumption:* $F_2 \in \mathcal{F}_{cc}$.
- *Initialization: Define a family of vectors* $w^x$ *defined for all* $i = 1, \ldots, \dim(X_2)$ *by*

$$w_i^x = \begin{cases} 1, & y_i \leq r_i \\ \frac{(v^x)_i}{t'_i(y_i)}, & y_i > r_i \end{cases}, \quad v^x \in \bar{\partial} F_2(y)\ , \quad y = G(x), \quad x \in X$$

*and starting point* $x^0 \in X$ *with* $F(x^0) < \infty$.

- *Iterations* $(k \geq 0)$*: Update*

$$x^{k+1} = \arg\min_{x \in X} F_1(x) + \left\langle w^{x^k}, T_{F_2}(G(x)) \right\rangle \tag{9}$$

As we already have shown, the functions $t_j$ majorize corresponding $f_j$. Weighting the functions with $w_i^x$ does not remove the majorization property. More precisely, if $v_i \in \bar{\partial} f_i(y_i^0)$, $w_i = v_i \cdot (t'_i(y_i^0))^{-1}$, then $w_i t_i(y_i) + f_i(y_i^0) - w_i t_i(y_i^0) \geq f(y_i)$ for all $y_i$.

## 5.3 Iteratively reweighted Huber algorithm

Consider the same class of functions $\mathcal{F}_{cc}$ as in the previous subsection. In practice it is beneficial when the majorizing function has simple analytic form. This may not be the case for tight convex majorizers introduced above. However, a wide class of functions can be majorized with help of the Huber function, which is defined as:

$$h_\varepsilon(\|x\|_2) = \begin{cases} \frac{1}{2\varepsilon}\|x\|_2^2, & \text{if } \|x\|_2 \le \varepsilon \\ \|x\|_2 - \frac{\varepsilon}{2}, & \text{otherwise.} \end{cases} \tag{10}$$

We define $H_\varepsilon(y) := (h_\varepsilon(y_1), \ldots, h_\varepsilon(y_K))$, which applies (10) coordinate-wise. Supposing that $F_2$ is differentiable on an open superset of $G(X)$, we can formulate the following algorithm:

**Algorithm 5** (Iteratively reweighted Huber algorithm: IRHuber)**.**

- *Assumption:* $F_2 \in \mathcal{F}_{cc}$.
- *Initialization: Define a family of vectors* $w^x$ *defined for all* $i = 1, \ldots, \dim(X_2)$ *by*

$$w_i^x = \frac{(\nabla F_2(y))_i}{h'_\varepsilon(y_i)}, \quad y = G(x), \quad x \in X$$

*and starting point* $x^0 \in X$ *with* $F(x^0) < \infty$.

- *Iterations* $(k \ge 0)$*: Update*

$$x^{k+1} = \arg\min_{x \in X} F_1(x) + \left\langle w^{x^k}, H_\varepsilon(G(x)) \right\rangle \tag{11}$$

**Example 4.** Consider the optimization problem

$$\min_{u \in X} \lambda\|u - f\|_1 + \tfrac{1}{2}\sum_i \log(1 + |Du|_i^2)\,,$$

where the convention for $|Du|_i$ is as in Example 3. The only difference to Example 3 is the square in the second term. However, as mentioned already, this makes IRL1 an unsuitable choice (see Remark 3). The term $F_2(G(x)) = \frac{1}{2}\sum_i \log(1 + |Du|_i^2)$ with $G(x) = |Du|_i$ is better approximated using the Huber function, which is quadratic close to 0. Vice verse, approximating the function from Example 3 with the Huber function is also a bad choice.

Obviously, $F_2$ is smooth and belongs to the class $\mathcal{F}_{cc}$. In order to write down the surrogate function we need to calculate the derivative of $F_2$. For all $i$, it is $(\nabla F_2(y))_i = y_i/(1 + y_i^2)$. The weights are chosen such that the surrogate function has the same slope as $\nabla F_2$ at $u^k$. As the Huber function has the derivative $(\nabla H_\varepsilon(y))_i = y_i/\varepsilon$, if $|y_i| \le \varepsilon$, and $(\nabla H_\varepsilon(y))_i = y_i/|y_i|$ otherwise, the weight vector $w^{u^k}$ is inferred as

$$w_i^{u^k} = \frac{\max\{\varepsilon, |\nabla u^k|_i\}}{1 + |\nabla u^k|_i^2}\,.$$

**Remark 3.** Within our framework there are different ways to approximate the function $F_2(G(x)) = \log(1 + |x|^2)$. We consider this in 1D here. The option we used in the preceding example corresponds to setting $G(x) = |x|$, $F_2(y) = \log(1 + y^2)$ and approximating $F_2(y)$ using the Huber function. However, we could also set $G(x) = |x|^2$, $F_2(y) = \log(1 + y)$ and approximate $F_2(y)$ as in the IRL1 algorithm. Then, the (convex) surrogate function to be minimized in the IRL1 algorithm is $F_1(x) + w^{x^k}G(x) = F_1(x) + w^{x^k}|x|^2$ and the approximation is by a quadratic function and hence worse than the approximation with the Huber function. This choice of $G$ and $F_2$ corresponds to the well-known iteratively reweighted least squares algorithm (IRLS), which we will recap in Subsection 5.4.

A natural question arises: which of the two interpretations of the problem leads to better results. We argue that setting $G(x) = |x|$, $F_2(y) = \log(1 + y^2)$ is better. In this case we can approximate $F_2(y)$ by the Huber function $H_\varepsilon(y)$, and the corresponding approximation for $G(x) = |x|^2$, $F_2(y)$ would be $H_\varepsilon(\sqrt{y})$. However, $H_\varepsilon(\sqrt{y})$ is non-convex and therefore not feasible for our iteratively reweighted convex method

(Method 2). This suggests to always choose $F_2$ and $G$ such that $G$ is as close as possible to $|x|$ (i.e. 'as non-convex as possible'). In this case $F_2^{x^k} \circ G$ can better approximate $F_2 \circ G$. For instance, IRHuber is a better approximation than IRLS.

As the majorization property of IRHuber is not immediately clear in this setup, we prove a general condition under which it holds and verify it for the preceding example.

**Proposition 3.** *Suppose $f\colon X \to \mathbb{R}$ and $m\colon X \to \mathbb{R}$, $X \subset \mathbb{R}$ open, are continuously differentiable non-decreasing functions and there exists a non-increasing function $r\colon \mathbb{R} \to \mathbb{R}_+$ such that $f'(x) = r(x)\, m'(x)$. Then for every $x_0 \in \mathbb{R}$ the function $m_{x_0}(x) = r(x_0)\, m(x) + f(x_0) - r(x_0)\, m(x_0)$ majorizes the function $f$.*

*Proof.* Obviously, $f(x_0) = m_{x_0}(x_0)$ and $f'(x_0) = m'_{x_0}(x_0)$. We then have for $x > x_0$:

$$m_{x_0}(x) - f(x) = \int_{x_0}^{x} \left(m'_{x_0}(t) - f'(t)\right) dt = \int_{x_0}^{x} \left((r(x_0) - r(t))m'(t)\right) dt \geq 0.$$

Similarly, for $x < x_0$:

$$m_{x_0}(x) - f(x) = -\int_{x}^{x_0} \left(m'_{x_0}(t) - f'(t)\right) dt = -\int_{x_0}^{x} \left((r(x_0) - r(t))m'(t)\right) dt \geq 0.$$

□

We now apply this proposition to the special case $f(x) = \log(1 + \mu x^2)$, $m(x) = h_\varepsilon(x)$. Since both functions are symmetric, we only consider $x \geq 0$. We then have:

$$\begin{aligned} f'(x) &= \frac{2\mu x}{1 + \mu x^2}, \\ m'(x) &= \min\left(\frac{x}{\varepsilon}, 1\right) = \begin{cases} \frac{x}{\varepsilon}, & 0 \leq x \leq \varepsilon, \\ 1, & x > \varepsilon, \end{cases} \\ r(x) &= \frac{f'(x)}{m'(x)} = 2\mu \, \frac{\max(x, \varepsilon)}{1 + \mu x^2}, \\ r'(x) &= 2\mu \begin{cases} -\frac{2\mu\varepsilon x}{(1+\mu x^2)^2}, & 0 \leq x \leq \varepsilon, \\ \frac{1-\mu x^2}{(1+\mu x^2)^2}, & x > \varepsilon. \end{cases} \end{aligned}$$

Obviously, $r$ is non-increasing as soon as $\varepsilon \geq \frac{1}{\sqrt{\mu}}$.

## 5.4 Iteratively reweighted least squares algorithm

The well-known IRLS Algorithm does also arise as a special case of Method 2. We present it in Algorithm 6 using our notation. Obviously, it is applicable at least to the same class of problems as Algorithm 5. Thus, the majorization property is clear.

**Algorithm 6** (Iteratively reweighted least squares algorithm: IRLS)**.**

- *Assumption:* $F_2 \in \mathcal{F}_{cc}$.
- *Initialization: Define a family of vectors* $w^x$ *defined for all* $i = 1, \dots, \dim(X_2)$ *by*

$$w_i^x = \frac{(\nabla F_2(y))_i}{y_i}, \quad y = G(x), \quad x \in X$$

*and starting point* $x^0 \in X$ *with* $F(x^0) < \infty$.

- *Iterations* $(k \geq 0)$*: Update*

$$x^{k+1} = \arg\min_{x \in X} F_1(x) + \left\langle w^{x^k}, \tfrac{1}{2}(G(x))^2 \right\rangle, \tag{12}$$

*where the square is to be understood coordinate-wise.*

**Example 5.** Consider Example 4. Using the IRLS algorithm the weight vector $w^{u^k}$ is given by

$$w_i^{u^k} = \frac{1}{1 + |Du^k|_i^2}.$$

However, the quadratic function is a worse approximation of the non-convex norm than the Huber function. We hence expect IRHuber to outperform IRLS.

## 5.5 Convergence analysis

Throughout the whole convergence analysis, let $(x^k)_{k \in \mathbb{N}}$ be a sequence generated by Method (1). We also always suppose that the functions $F$, $F_1$, $F_2$, $G$ fulfill the conditions stated in Section 3. We make frequent use of the tools from variational analysis presented in the Appendix. In addition, from now on we assume $F$ to be coercive, i.e., $F(x) \to \infty$, whenever $\|x\| \to \infty$.

**Proposition 4.** *Let* $F$ *be coercive, then the sequence* $(x^k)_{k \in \mathbb{N}}$ *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

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

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

**Additional assumptions.** In order to prove convergence for the whole sequence $(x^k)_{k \in \mathbb{N}}$, two additional assumptions are required. Let us discuss them.

1. We assume that $F_2$ has locally Lipschitz continuous gradient (see Definition 5 and the subsequent comment) on a compact set $B$ containing all the points $x^k$ and that $F_2^x$ have globally Lipschitz continuous gradients on $B$ for all $x \in X$ with a common Lipschitz constant $\widetilde{L} \geq 0$. This assumption is less restrictive as it seems to be. Many non-smooth functions may be written as a sum of a function with locally Lipschitz gradient and a convex function. Then, the convex part may be shifted to $F_1$. This class of functions was for example considered in [3].

2. $F_1 + F_2^{x^k} \circ G$ must be strongly convex (see Definition 6) with a constant independent of $k$. Otherwise, the sum $F_1 + F_2^{x^k} \circ G$ can have a plateau as local minimum, i.e., there is no unique minimizer. This

happens for example when $F_1(x) = |x-1|$ and $F_2^{x^k}(G(x)) = |x|$ for all $x^k \in [0,1]$. Our algorithm then has to choose from multiple equally good solutions and hence may not converge. One standard way to resolve this problem is to add a proximity term $c\|x - x^k\|_2^2$ to the convex surrogate problem (4) with arbitrarily small $c > 0$. This makes the surrogate problem strongly convex and makes the algorithm converge to one solution from the plateau.

A technical assumption we make from now on is that $F_2$ and $F_2^x$ for all $x \in X$ are defined on open sets comprising $G(X)$ and continuously differentiable on $G(X)$. In all practical cases this is clearly fulfilled. The following properties then hold:

**Lemma 1.** *Under the aforementioned conditions, it holds for all $\bar{x} \in X$*

1. *and for all $x \in X$*

$$\partial(F_2^{\bar{x}} \circ G)(x) = \partial \langle y, G\rangle (x) \text{ with } y = \nabla F_2^{\bar{x}}(x)\,,$$
$$\partial(F_2 \circ G)(x) = \partial \langle y, G\rangle (x) \text{ with } y = \nabla F_2(x)\,,$$

2. *and for all $x \in \operatorname{dom} F_1$ and all $x \in X$*

$$\partial(F_1 + F_2^{\bar{x}} \circ G)(x) = \partial F_1(x) + \partial(F_2^{\bar{x}} \circ G)(x)\,,$$
$$\partial(F_1 + F_2 \circ G)(x) = \partial F_1(x) + \partial(F_2 \circ G)(x)\,.$$

*Proof.* We verify the second equality for both items. The first one follows analogously.

1. Since $F_2$ is continuously differentiable on an open set containing $G(X)$, for $x \in X$ it is $\partial^\infty F_2(G(x)) = \{0\}$ [76, Ex. 8.8]. Continuous differentiability also yields regularity of $F_2$ at $G(x)$ for $x \in X$ [76, Ex. 7.28]. By assumption $F_2$ is coordinate-wise non-decreasing, which implies that $\langle y, G\rangle (x)$ with $y = \nabla F_2(x)$ is a lsc., convex function. As a consequence, $\langle y, G\rangle (x)$ is regular at $x \in X$, which verifies the conditions for equality in Proposition 10. As a side product, $F_2 \circ G$ is regular for all $x \in X$.
2. Convexity of $G$ implies its local Lipschitz continuity [76, Ex. 9.14] and, hence, also local Lipschitz continuity of $F_2 \circ G$. Therefore, $\partial^\infty(F_2 \circ G)(x) = \{0\}$ (see Proposition 8), which together with convexity of $F_1$ (hence $\partial F_1(x) = \widehat{\partial} F_1(x)$) and Clarke regularity of $F_2 \circ G$ at $x$ (see first point in this proof) ensures $\partial F_1(x) + \partial(F_2 \circ G)(x) = \partial(F_1 + F_2 \circ G)(x)$ (see Proposition 9).

□

**Proposition 5.** *Let $B$ be a bounded set containing all $x^k$. Let $F_2$ have locally Lipschitz continuous gradient on $B$ and let $F_2^x$ have globally Lipschitz continuous gradients on $B$ for all $x \in X$ with a common Lipschitz constant $\widetilde{L} \geq 0$. Let also $F_1 + F_2^{x^k} \circ G$ be strongly convex with convexity parameter $\mu > 0$ for all $x^k \in X$. Then, the following holds*

1. *$F(x^{k+1}) \leq F(x^k) - \frac{\mu}{2}\|x^k - x^{k+1}\|_2^2$ for all $k \in \mathbb{N}$,*
2. *there exists $C > 0$ such that for all $k \in \mathbb{N}$ there exists $\xi^{k+1} \in \partial F(x^{k+1})$ fulfilling*

$$\|\xi^{k+1}\|_2 \leq C\|x^{k+1} - x^k\|_2\,,$$

3. *and for any converging subsequence $(x^{k_j})_{j\in\mathbb{N}}$ with $\bar{x} := \lim_{j\to\infty} x^{k_j}$ holds $F(x^{k_j}) \to F(\bar{x})$ as $j \to \infty$.*

*Proof.*

1. The strong convexity of $F_1 + F_2^{x^k} \circ G$ provides for all $\xi_1 \in \partial F_1(x^{k+1})$ and $\xi_2^k \in \partial(F_2^{x^k} \circ G)(x^{k+1})$ the inequality

$$\begin{aligned} &F_1(x^{k+1}) - F_1(x^k) + F_2^{x^k}(G(x^{k+1})) - F_2^{x^k}(G(x^k))) \\ &\qquad\qquad \leq \left\langle \xi_1, x^k - x^{k+1} \right\rangle + \left\langle \xi_2^k, x^k - x^{k+1} \right\rangle - \frac{\mu}{2}\|x^{k+1} - x^k\|_2^2 \,, \end{aligned}$$

   As $x^{k+1}$ is a minimizer of (4) and thanks to Lemma 1, we can choose $\xi_1 + \xi_2^k = 0 \in \partial(F_1 + F_2^{x^k} \circ G)(x^{k+1})$. Using $F_2(G(x^{k+1})) \leq F_2^{x^k}(G(x^{k+1}))$ and $F_2(G(x^k)) = F_2^{x^k}(G(x^k))$, we conclude this part of the proof.

2. Local Lipschitz continuity of $G$ (which follows from its convexity) and the gradient of $F_2$ provides their global Lipschitz continuity on $B$. We denote the corresponding Lipschitz constants by $L_G$ and $L$ respectively.

   Using Lemma 1, we can select $\xi_1 \in \partial F_1(x^{k+1})$ and $\xi_2^k \in \partial(F_2^{x^k} \circ G)(x^{k+1})$ such that

$$\xi_1 + \xi_2^k = 0 \in \partial(F_1 + F_2^{x^k} \circ G)(x^{k+1}) = \partial F_1(x^{k+1}) + \partial(F_2^{x^k} \circ G)(x^{k+1}) \,.$$

   Then, for all $\xi_2 \in \partial(F_2 \circ G)(x^{k+1})$ it holds

$$\|\xi_1 + \xi_2\|_2 = \|\xi_1 + \xi_2 - \xi_1 - \xi_2^k\|_2 = \|\xi_2 - \xi_2^k\|_2 \,. \tag{13}$$

   Using the chain rule from Proposition 10 and Lemma 1, we have (define $y^k := \nabla F_2^{x^k}(G(x^{k+1})))$

$$\partial(F_2^{x^k} \circ G)(x^{k+1}) = \partial \left\langle y^k, G \right\rangle(x) = \sum_i \partial(y_i^k G_i)(x^{k+1}) = \sum_i y_i^k \partial G_i(x^{k+1})$$

   and, thus, we can decompose $\xi_2^k = \sum_i y_i^k \eta_i$ with $\eta_i \in \partial G_i(x^{k+1})$. We then define $\xi_2 := \sum_i y_i \eta_i$, where $y := \nabla F_2(G(x^{k+1}))$. The combination of both decompositions together with the Lipschitz continuity of $G$ and [76, Prop. 9.24] yields

$$\|\xi_2 - \xi_2^k\|_2 = \|\sum_i (y - y^k)_i \eta_i\|_2 \leq L_G \|y - y^k\|_2 \,. \tag{14}$$

   Now, using (13) and (14), the equality $\nabla F_2(G(x^k)) = \nabla F_2^{x^k}(G(x^k))$, the Lipschitz continuity of $\nabla F_2$ and $\nabla F_2^{x^k}$ and noting that $\xi_1 + \xi_2 \in \partial F(x^{k+1})$, the following estimation concludes this part of the proof:

$$\begin{aligned} \|\xi_1 + \xi_2\|_2 &\leq L_G \|y - \nabla F_2(G(x^k)) + \nabla F_2^{x^k}(G(x^k)) - y^k\|_2 \\ &\leq (L + \widetilde{L}) L_G \|G(x^{k+1}) - G(x^k)\|_2 \\ &\leq (L + \widetilde{L}) L_G^2 \|x^{k+1} - x^k\|_2 \,, \end{aligned}$$

   where the last transition follows from the Lipschitz continuity of $G$.

3. Let $(x^{k_j})_{j\in\mathbb{N}}$ be a converging subsequence of $(x^k)_{k\in\mathbb{N}}$. Define the sequences $(\xi_1^{k_j})_{j\in\mathbb{N}}$ and $(\xi_2^{k_j})_{j\in\mathbb{N}}$ by $0 = \xi_1^{k_j} + \xi_2^{k_j} \in \partial F_1(x^{k_j}) + \partial(F_2^{x^{k_j-1}} \circ G)(x^{k_j})$, which by Lemma 1 coincides with $\partial(F_1 + F_2^{x^{k_j-1}} \circ G)(x^{k_j})$. Due to the local Lipschitz continuity of $F_2^{x^{k_j-1}} \circ G$ Proposition 8 implies $x \mapsto \partial(F_2^{x^{k_j-1}} \circ G)(x)$ bounded, and therefore, the sequence $(\xi_1^{k_j})_{j\in\mathbb{N}}$ is bounded and $\lim_{j\to\infty} \left\langle \xi_1^{k_j}, \bar{x} - x^{k_j} \right\rangle = 0$. Using this, $F$ lsc, $F_1$ convex, and $F_2 \circ G$ locally Lipschitz continuous, the following chain of inequalities concludes the proof (all limits are considered for $j \to \infty$):

$$\begin{aligned} F(\bar{x}) &\leq \liminf F(x^{k_j}) \leq \limsup F(x^{k_j}) \\ &\leq \limsup F_1(x^{k_j}) + \limsup F_2(G(x^{k_j})) \\ &= \limsup F_1(x^{k_j}) + \lim \left\langle \xi_1^{k_j}, \bar{x} - x^{k_j} \right\rangle + F_2(G(\bar{x})) \\ &= \limsup \Big( F_1(x^{k_j}) + \left\langle \xi_1^{k_j}, \bar{x} - x^{k_j} \right\rangle \Big) + F_2(G(\bar{x})) \\ &\leq F_1(\bar{x}) + F_2(G(\bar{x})) = F(\bar{x}) \,. \end{aligned}$$

□

In [6], an abstract convergence result for descent methods for semi-algebraic and tame problems is proved. We recap the result in Theorem 3 in the appendix. The notion of semi-algebraic functions and the KL property, which is central to the theorem, are introduced it in the Appendix (Definition 7 and 8). In the following theorem, we benefit from their convergence analysis by simply proving our algorithm to satisfy their assumptions.

**Theorem 2.** *Let the assumptions be as in Proposition 5. Let the sequence $(x^k)_{k\in\mathbb{N}}$ be generated by Method 1. If $F$ has the Kurdyka-Lojasiewicz property at the cluster point $x^* := \lim_{j\to\infty} x^{k_j}$, then the sequence $(x^k)_{k\in\mathbb{N}}$ converges to $x^* \in X$ as $k \to \infty$ and $x^*$ is a critical point of $F$. Furthermore, the sequences $(x^k)_{k\in\mathbb{N}}$ has finite length*

$$\sum_{k=0}^{\infty} \|x^k - x^{k+1}\|_2 < \infty\,.$$

*Proof.* The results of Proposition 4 and Proposition 5 are exactly the requirements of Theorem 3 (which is copied from [6, Theorem 2.9]). Applying this result proves the theorem. □

# 6 Prototypes for computer vision applications

Many computer vision examples involve a linear operator in order to enforce spatial regularity of the solution. For example, this can be achieved using the gradient operator. We consider the prototype of inverse problems in computer vision

$$\min_{u\in X} \|Au - g\|_q^q + F_2(\tilde{G}(Ku))\,, \tag{15}$$

where $q \in \{1, 2\}$ and $K\colon X \to X$ may be any continuous linear operator (for example, gradient operator). Since in computer vision mostly the optimization variable, which often is an image, is denoted by $u$, we adapt this notation from now on. In the original formulation (2), it is $G = \tilde{G} \circ K$. Here, we further assume $\tilde{G}(0) = 0$, and $\tilde{G}(u)_i \geq 0$. A common choice for $K$ is the gradient operator $D = (D_x^\top, D_y^\top)^\top$ ($D_x$ is a matrix implementing forward differences in $x$-direction; analogue for $D_y$) and for $\tilde{G}$ the length of a vector $\tilde{G}((D_x u)_i, (D_y u)_i) = \sqrt{(D_x u)_i^2 + (D_y u)_i^2}$. In the first term of (15), called the *data-term*, we denote by $A\colon X \to X_1$ a continuous linear operator and by $X_1$ a finite dimensional real vector space. This linear operator maps into a space, where measurements $g \in X_1$ are taken. The second term is denoted the *regularization-term*. Non-convex regularization functions $F_2$ suitable for computer vision applications were already shown in Figure 1. The prototypes for the function $F_2 \circ G$ are

$$u \mapsto \frac{1}{p}\|\tilde{G}(Ku)\|_{p,\varepsilon}^p := \frac{1}{p}\sum_i (\tilde{G}_i(Ku) + \varepsilon)^p, \quad \varepsilon > 0, p \in (0, 1] \tag{16}$$

$$u \mapsto \frac{1}{\mu}\log(1 + \mu\tilde{G}(Ku)) := \frac{1}{\mu}\sum_i \log(1 + \mu\tilde{G}_i(Ku)), \quad \mu > 0 \tag{17}$$

$$u \mapsto \frac{1}{2\mu}\log(1 + \mu\tilde{G}(Ku)^2) := \frac{1}{2\mu}\sum_i \log(1 + \mu(\tilde{G}_i(Ku))^2), \quad \mu > 0\,. \tag{18}$$

The first and the second functions $F_2$ are concave and non-decreasing on $G(X)$ but non-differentiable, and the third is non-decreasing and differentiable. These functions clearly fulfill differentiability and Lipschitz continuity conditions required for our convergence analysis to hold. We now show that KL-property also holds:

**Proposition 6.** *Let $F_2$ be one of the prototypes* (16), (17)*, or* (18)*, and let $\tilde{G}$ be semi-algebraic. Then, the function $F(u) = \|Au - g\|_q^q + F_2(\tilde{G}(Ku))$ is a KL-function.*

*Proof.* As $\tilde{G}$, $K$, and $\|Au-g\|_q^q$ are semi-algebraic (simple compositions of semi-algebraic functions), it is enough to verify that $F_2$ is definable in an o-minimal structure. However, thanks to the log–exp structure [84, 32], this fact is also clear for Prototypes (18), (17), and (16) (note that $(u+\varepsilon)^p = \exp(p\log(u+\varepsilon))$, $u \geq 0$). Then, [15, Theorem 14] implies that $F$ has the KL-property at any stationary point. □

## 6.1 Total generalized variation regularization

Opposed to TV-regularization which is used very frequently and can be seen as basic knowledge total generalized variation (TGV) regularization was introduced only recently [17]. The following introduction to TGV will be given in the continuous setting. For details we refer to [17].

TGV generalizes TV based on the dual formulation incorporating the space of $k$-tensors

$$\begin{aligned}\mathcal{T}^k(\mathbb{R}^d) &:= \{\xi\colon \mathbb{R}^d\times\ldots\times\mathbb{R}^d \to \mathbb{R} : \xi \text{ is } k\text{-linear}\}\\ \mathrm{Sym}^k(\mathbb{R}^d) &:= \{\xi\colon \mathbb{R}^d\times\ldots\times\mathbb{R}^d \to \mathbb{R} : \xi \text{ is } k\text{-linear and symmetric}\}\,.\end{aligned}$$

Let $\Omega\subset\mathbb{R}^2$ be the image domain and $u\colon\Omega\to\mathbb{R}$ be a function, then the TGV semi-norm of order $k\geq 1$ with smoothness parameter $\alpha=(\alpha_0,\ldots,\alpha_{k-1})$ is defined by

$$TGV_k^\alpha(u) := \sup\left\{\int_\Omega u \operatorname{div}^k\varphi\,dx\,\middle|\,\varphi\in\mathcal{C}_c^k(\Omega,\mathrm{Sym}^k(\mathbb{R}^2)),\|\operatorname{div}^l\varphi\|_\infty\leq\alpha_l, l=0,\ldots,k-1\right\},$$

where $\mathcal{C}_c^k(\Omega,\mathrm{Sym}^k(\mathbb{R}^2))$ denotes the space of continuously differentiable symmetric $k$-tensor fields with compact support in $\Omega$, $\operatorname{div}^k$ the generalization of the divergence operator to these tensor fields. For $k=1$ the definition of TGV reduces to the dual formulation of the TV semi-norm.

Usually a primal formulation yields more intuition about a new concept. As in this paper, we are only interested in $TGV_2^\alpha(u)$ we specify the order $k=2$ in the following. Applying the Legendre-Fenchel transform yields

$$TGV_2^\alpha(u) = \inf_{u_1\in C^1(\overline{\Omega},\mathrm{Sym}(\mathbb{R}^2))} \alpha_1\|Du-u_1\|_1 + \alpha_0\|\mathcal{E}(u_1)\|_1\,,$$

where $\mathcal{E}$ denotes the symmetrized gradient operator $\mathcal{E}(u_1) = (Du_1+Du_1^\top)/2$, which is a $2\times 2$-matrix. There is also an asymmetric version of TGV defined in the primal formulation as

$$asymTGV_2^\alpha(u) = \inf_{u_1\in C^1(\overline{\Omega},\mathcal{T}(\mathbb{R}^2))} \alpha_1\|Du-u_1\|_1 + \alpha_0\|Du_1\|_1\,.$$

Note, that the primal formulation of semi-norm $TGV_2^\alpha$ itself is stated as a minimization problem. However, when optimizing a function with $TGV_2^\alpha$ as a regularizer, we consider the single minimization problem in the variables $u$ and $u_1$.

The main property of $TGV_2^\alpha$ is the ability to reconstruct piecewise affine functions without penalty. This makes $TGV_2^\alpha$ favorable compared to TV, which can only reconstruct piecewise constant functions. Considering the primal formulation, the intuition about the behavior of $TGV_2^\alpha$ can be explained as follows. Note that $u_1$ may be constant without increasing the norm. Then, $u$ is allowed to be linear because $Du$ may be constant (the constant of $u_1$) without increasing the TGV semi-norm.

# 7 Experimental analysis

## 7.1 Implementation details

The convex subproblems arising for the non-convex optimization problems that are considered in the following can be solved efficiently, see for example [61, 8, 23]. If not stated differently, we use the respective optimal algorithm from [23]. It has proved optimal convergence rate: $O(1/e^n)$ when $F_1$ and $F_2^*$ (convex conjugate of $F_2$) are uniformly convex, or when $F_1$ is uniformly convex and $F_2$ has Lipschitz continuous gradient, $O(1/n^2)$ when $F_1$ or $F_2$ is uniformly convex and $O(1/n)$ for the general case.

Here, we focus on the (outer) non-convex problem. Let $(u^{k,l})$ be the sequence generated by Method 2 (or Method 1), where the index $l$ refers to the inner iterations for solving the convex problem, and $k$ to the outer iterations. Proposition 1, which proves $(F(u^{k,0}))$ to be monotonically decreasing, provides a natural stopping criterion for the inner and outer problem. We verify every 10th inner iteration and stop as soon as

$$F(u^{k,l}) < F(u^{k,0}) \quad \text{or} \quad l > m_i, \tag{19}$$

where $m_i$ is the maximal number of inner iterations. For a fixed $k$, let $l_k$ be the number of iterations required to satisfy the inner stopping criterion (19). Then, outer iterations are stopped when

$$\frac{F(u^{k,0}) - F(u^{k+1,0})}{F(u^{0,0})} < \tau \quad \text{or} \quad \sum_{i=0}^{k} l_i > m_o, \tag{20}$$

where $\tau$ is a threshold defining the desired accuracy and $m_o$ the maximal number of iterations. The difference in (20) is normalized by the initial function value to be invariant to a scaling of the energy.

In order to obtain a guarantee for a converging sequence of function values checking the decent property is required. However, throughout the experiments we observed that a fixed number of 10 iterations is a good choice and we can omit computing the energy.

**Remark 4.** As long as we can guarantee that the energy decreases we can expect a converging sequence of function values. However, if the subproblem is not solved exactly, the convergence properties from Theorem 2 are partially lost. The convergence theorem allows for inexact descent methods, i.e., it allows for some errors in the evaluation of the subproblem. However the granted quantity of the error is not addressed in the theorem. Therefore, we focus on the convergence of the energy values.

## 7.2 Competing method

We compare our algorithm against another recently proposed algorithm for non-convex optimization: *iPiano* [67]. It is a forward-backward splitting algorithm incorporating an inertial force. We compare against iPiano, because it has proved to be very efficient and it is applicable to similar problems as considered in this paper, namely to problems that can be decomposed as a sum of a (simple) convex function and a function with Lipschitz continuous gradient. Moreover, iPiano finds special cases in the NIPS algorithm [78] when the inertial term is turned off, in the Heavy-ball method for differentiable non-convex problems [86], or in the well-known gradient projection algorithm. Therefore, actually our comparison is against several algorithms.

Assuming that our $F_2 \circ G$ has Lipschitz continuous gradient with constant $L > 0$, the update scheme of iPiano using our notation can be written as

$$u^{n+1} = (I + \alpha\partial F_1)^{-1}(u^n - \alpha\nabla(F_2 \circ G)(u^n) + \beta(u^n - u^{n-1}))\,. \tag{21}$$

Due to the smoothness assumption to $F_2 \circ G$ in this algorithm, when comparing to the proposed IRL1-algorithm the non-differentiable points must be smoothed.

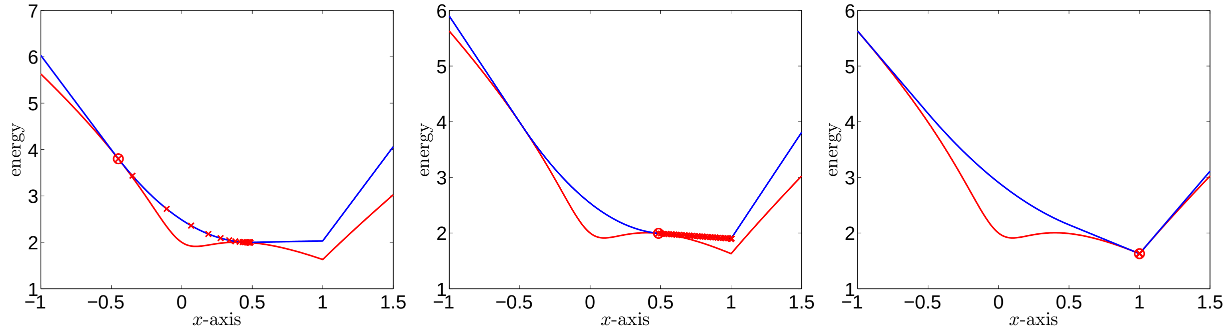
## 7.3 Analysis of local minima

In this part, we experimentally study the sensitivity of our algorithm with respect to local stationary points.

### 7.3.1 A one dimensional example

Here, we show that the proposed algorithm has the ability to avoid local minima. We consider the model problem (see red function in Figure 3)

$$\min_{u\in\mathbb{R}} \ \lambda|u - f| + \frac{1}{2}\log(1 + \mu|u|^2)\,,$$



(a) outer iteration 1

(b) outer iteration 2

(c) outer iteration 3

Figure 3: Three (outer) iteration steps of the proposed algorithm to minimize the red non-convex function. In each plot a few steps of the solver for the (inner) convex problem (minimization of the blue function) are visualized. The red circle shows the point, in which the original function is approximated. Though there is a local minimum close to the starting point, the algorithm jumps over this and finds the global optimum.

| $\mu = 1$ | $\lambda = 0.10$ | $\lambda = 0.50$ | $\lambda = 1.00$ | $\lambda = 2.00$ | $\lambda = 5.00$ |
|---|---|---|---|---|---|
| min. energy | 92.39 | 369.69 | 592.51 | 683.67 | 683.67 |
| PrimalDual-IRHuber | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| iPiano, $\beta = 0.7$ | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| $\mu = 50$ | $\lambda = 0.10$ | $\lambda = 0.50$ | $\lambda = 1.00$ | $\lambda = 2.00$ | $\lambda = 5.00$ |
| min. energy | 124.80 | 559.76 | 1060.89 | 1984.31 | 5097.96 |
| PrimalDual-IRHuber | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| iPiano, $\beta = 0.7$ | **1.0000** | **1.0000** | **1.0000** | **1.0000** | 1.0006 |
| $\mu = 100$ | $\lambda = 0.10$ | $\lambda = 0.50$ | $\lambda = 1.00$ | $\lambda = 2.00$ | $\lambda = 5.00$ |
| min. energy | 130.96 | 586.67 | 1118.07 | 2101.04 | 5945.98 |
| PrimalDual-IRHuber | **1.0000** | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| iPiano, $\beta = 0.7$ | **1.0000** | **1.0000** | **1.0000** | 1.0002 | 1.0050 |
| $\mu = 250$ | $\lambda = 0.10$ | $\lambda = 0.50$ | $\lambda = 1.00$ | $\lambda = 2.00$ | $\lambda = 5.00$ |
| min. energy | 140.87 | 623.65 | 1189.83 | 2255.34 | 6986.61 |
| PrimalDual-IRHuber | **1.0000** | 1.0007 | 1.0007 | **1.0000** | **1.0000** |
| iPiano, $\beta = 0.7$ | **1.0000** | **1.0000** | **1.0000** | 1.0014 | 1.0145 |

Table 1: Comparison of the final energy for our IRHuber algorithm compared to iPiano for the problem (22) and different parameter settings with maximal 50000 iterations. *min. energy* is the minimal final energy value among the four methods. The other values describe the multiplication factor to this minimal energy. In most experiments, IRHuber finds the lowest energy.

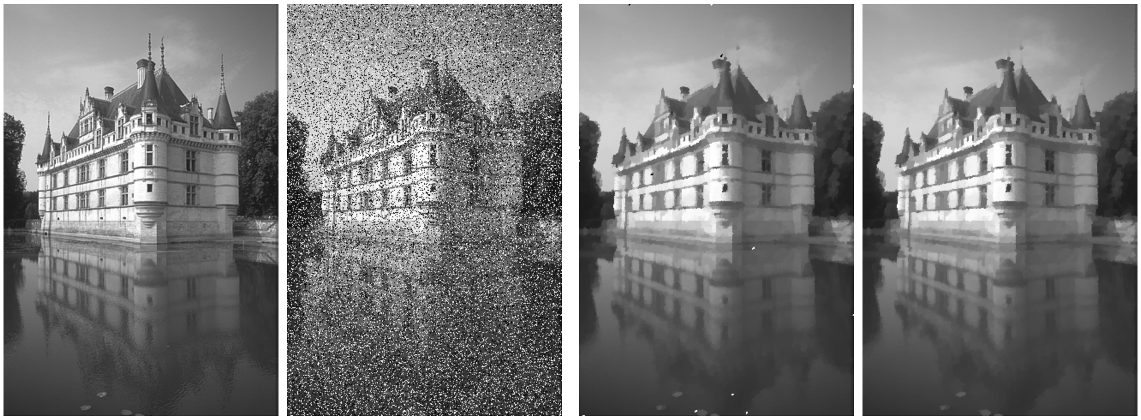
where $f = 1$, $\mu = 25$, $\lambda = 2$. As it is $F_2(|u|) = \frac{1}{2}\log(1 + \mu|u|^2)$, we use the iteratively reweighted Huber Algorithm 5 ($\varepsilon = 1$) to find a minimum of this function. Figure 3 shows three outer iterations of IRHuber initialized at $u^0 = -0.45$. Depending on the initialization different local optima are reached. Initializing $u^0 = \pm 0.4$ the local maximum at $u = 0.4$ is found, for $u^0 \in (-0.4, 0.4)$ the left local optimum is the solution, and initializing with $u^0 \in (-\infty, -0.4) \cup (0.4, \infty)$ the global optimum is found. Although the algorithm can also converge to a local maximum, this is rarely the case. When we initialize at $u^0 \in (-0.4, 0.4)$ the algorithm is already trapped to the left local minimum. However, different to many other method it does not necessarily converge to the nearest local minimum.

### 7.3.2 A high dimensional example

In image processing, optimization problems usually have a very high dimensionality and it is not possible to visualize the objective functions. Conclusions about whether the algorithm is attracted by local minima or whether it is "robust" against local minima can only be drawn indirectly. If the energy value (function value) corresponding to one algorithm is lower than with another, we conclude that the first algorithm has

| initialization | noisy image | zero | random | random with square of zero-valued pixel |
|---|---|---|---|---|
| initial energy | 45308.479 | 70803.569 | 116368.474 | 114674.008 |
| final energy | 23583.466 | 23575.354 | 23576.401 | 23576.01 |

Table 2: Initial and final energy values for the optimization problem 23 with $\lambda = 1$ and $\mu = 1$. Numerically, the result values slightly differ from each other. On average the difference is very small. Visually, different initializations yield very similar results. This suggests that our algorithm is robust towards the initialization in this experiment.



(a) ground truth and noisy image

(b) result for initialization with noisy image and zero image

Figure 4: Visualization of the experiment in (23). The results for two different initializations, the zero image and the noisy image, are shown in (b). As the corresponding energy values from Table 2 suggest the result images are visually close to each other.

found a better local minimum. This will be shown in the following example. In this experiment, we consider the problem

$$\min_{u \in \mathbb{R}^{6305}} \lambda \|u - f\|_1 + \frac{1}{2} \sum_i \log(1 + \mu |Du|_i^2) \tag{22}$$

and solve it using iPiano and our IRHuber ($\varepsilon = 1/\sqrt{\mu}$) method. For all methods we fix a maximum of 50000 iterations and use the break condition (20) with $\tau = 10^{-12}$. Table 1 confirms that our algorithm usually finds the lowest energy. The difference is more significant, the higher the values of $\lambda$ and $\mu$. For larger $\mu$ the "non-convexity" is stronger. For small $\lambda$ the optimal result is constant, i.e. $|Du|_i$ is small everywhere and lies in the convexity region of $\log(1+\mu y^2)$. Thus the non-convexity of the second term is of little importance.

### 7.3.3 Robustness towards the initialization

We fix $\lambda = 1$ and $\mu = 1$ and solve the optimization problem

$$\min_{u \in \mathbb{R}^N} \lambda \|u - f\|_1 + \sum_i \log(1 + \mu |Du|_i) \tag{23}$$

starting from different initializations $u^0$ using the iteratively reweighted $\ell_1$ algorithm (Algorithm 3). Here $N = 154401$. The noisy input image and the ground truth are shown in Figure 4(a). The energy values of the initialization and the final values are shown in Table 2. The energy values of the solutions slightly differ.

The maximal difference of energy values is between initializing with the noisy image and initializing with the zero image. The energy difference is approximately $d \approx 8.11$. Let us consider what it means per pixel on average and in the worst case. If we assume that this error is only caused by the data term, we can conclude

$$d = \sum_{i=1}^{N} |u_i - f_i| \geq N \min_i |u_i - f_i| \quad \Rightarrow \quad \min_i |u_i - f_i| \leq d/N \approx 5.25 \cdot 10^{-5}\,.$$

This means that the average (minimal) error per pixel is bounded by approximately $5.25 \cdot 10^{-5}$. Pixels that cause an error that is higher than the minimal one reduce the upper bound for the error for all other pixels. Considering the worst case only 8 pixel can have the maximal error of 1, which is the range of the gray values. On the other hand, if we assume the error is solely by the regularization term, it holds

$$d = \sum_{i=1}^{N} \log(1+|y_i|) \geq N \log(1 + \min_i |y_i|) \quad \Rightarrow \quad \min_i |y_i| \leq \exp(d/N) - 1 \approx 5.25 \cdot 10^{-5} \,.$$

Therefore, the energy difference could also be caused by an average (minimal) error for the gradient of maximal $5.25 \cdot 10^{-5}$. The worst case analysis shows that only $8.11/\log(2) \approx 11$ pixel can have an error in the gradient of 1. These numbers suggest a small difference between the two solutions. Figure 4(b) visualizes this difference. In this experiment, the final results are very similar. The error seems to be better reflected by the worst case analysis as it is concentrated on a few outliers.

Unfortunately, it is hard to generalize this observation. For input images with more noise or for different parameter settings the results can differ more depending on the initialization. The main problem is that the high dimensionality makes it hard to get an intuition about local minima and maxima. In the following experiments, we always initialize with the noisy image.

## 7.4 Numerical comparison

The existence of local minima and the missing information about the global optimum for non-convex optimization problems complicates the evaluation. For all the following experiments, we agree the following evaluation: We use the method that achieves the lowest energy value and run it for $10^6$ iterations; we use the solution $u^{10^6}$ to define $E^* := E(u^{10^6})$. Then, we use the relative distance to this "optimal" energy value and analyze the convergence of the sequence

$$\left( \frac{E(u^n) - E^*}{E(u^0) - E^*} \right)_{n \in \mathbb{N}} \,.$$

Note that, if the sequence does not convergence to $E^*$, then the sequence can still converge to another local optimum. As we choose $E^*$ such that it is minimal among the methods under consideration, a method that does not converge to $E^*$ only finds a higher energy.

**Iteratively reweighted Huber vs. iteratively reweighted tight convex.** First, we compare IRTight (Algorithm 4) vs. IRHuber (Algorithm 5 with $\varepsilon = 1/\sqrt{\mu}$ as suggested at the end of Subsection 5.3) for the optimization problem

$$\min_{u \in \mathbb{R}^{154401}} \frac{\lambda}{2} \|u - f\|_2^2 + \frac{1}{2\mu} \sum_i \log(1 + \mu |(Du)_i|^2) \,, \tag{24}$$

where $\lambda = 0.1$, $\mu = 800$, and $f \in \mathbb{R}^{154401}$ is the given noisy input image from Figure 6.

The convergence plot of the energy is shown in Figure 5. IRHuber converges faster in terms of the actual computation time than IRTight. This result is explained by the simple structure of the IRHuber surrogate function. The proximity operator that arises in the convex surrogate problem for the IRHuber can be solved analytically. For IRTight solving the proximity operator requires to find the zero of a cubic polynomial in a certain interval. For IRTight {1,5,10} this proximity operator is solved using Newton's method with a maximum of 1, 5 or 10 iterations and break condition for the maximal absolute difference of two successive iterates of $10^{-4}$. Closer to the optimal value the number of iterations required by Newton's method decreases due to the initialization. The analytic solution for the proximity operator needs always the same time. In terms of iterations, all methods perform equally well.

Thanks to the simple structure of IRHuber, it is more efficient for regularization problems involving terms $\log(1 + y^2)$. Therefore, in the following experimental comparison, we consider IRHuber only. However, we should keep in mind, that if the proximity operator in IRTight is easy to solve, it is a better approximation and convergences faster.

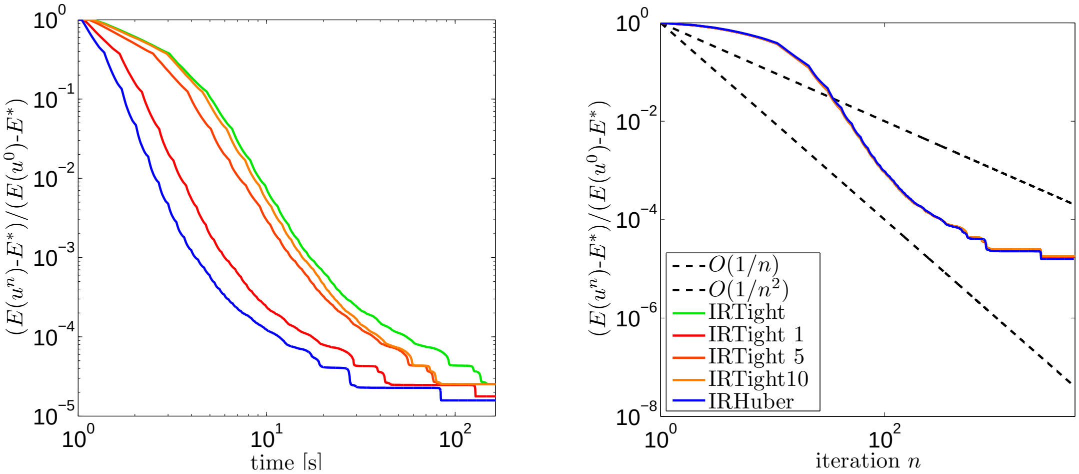
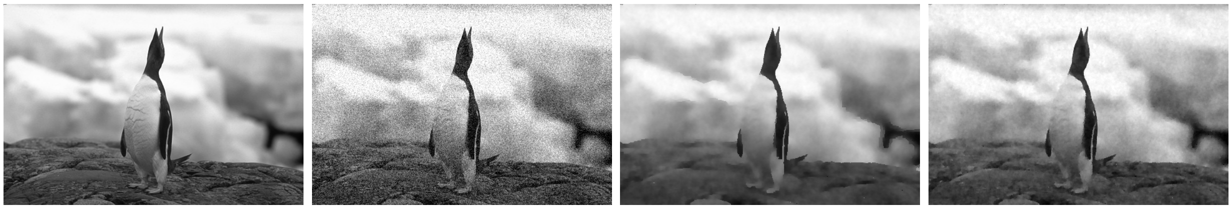


Figure 5: Comparison of the energy decrease for problem (24) between Algorithm 4 (IRTight) and Algorithm 5 (IRHuber). The legend is the same for both plots. The proximity operator arising in the convex optimization algorithm for IRTight is solved analytically or using 1, 5 or 10 Newton iterations. The proximal operator for IRHuber can be solved analytically. Therefore, in terms of runtime IRHuber is faster than IRTight, whereas in terms of iterations all methods perform equally well.



(a) ground truth (b) Gaussian noise added (c) (24) with $\lambda = 0.1$, $\mu = 800$ (d) (24) with $\lambda = 0.3$, $\mu = 250$

Figure 6: Visualization of the experiment (24).

**Iteratively reweighted Huber and reweighted least squares.** We evaluate our algorithm in terms of speed compared to iPiano (and its special case NIPS with $\beta = 0$). We consider the problem

$$\min_{u\in\mathbb{R}^{154401}} \frac{\lambda}{2}\|u-f\|_2^2 + \frac{1}{2\mu}\sum_i \log(1+\mu|Du|_i^2)\,, \tag{25}$$

where $\lambda = 0.3$, $\mu = 250$, and $f \in \mathbb{R}^{154401}$ is the given noisy input image from Figure 6 (see the same figure for the result image).

The Lipschitz constant required by iPiano is set to $L = 8$. Then, $\alpha = 2(1-\beta)/L$ is set according to the step size rules (see [67] for details). Using Algorithm 5 and Algorithm 6, the (convex) surrogate function is strongly convex and can be solved with linear convergence rate, which is optimal for this class of problems. Algorithm *iPiasco-IRHuber* solves the primal problem using iPiasco [66] and *D-iPiasco-IRHuber* solves the surrogate problem in the dual formulation. Analogously, *iPiasco-IRLS* solves the primal problem arising in Algorithm 6 and *D-iPiasco-IRLS* the dual problem. *CG-IRLS* solves the primal inner problem using conjugate gradient. In this experiment we do not show the result of solving the inner problem with the optimal primal dual algorithm [23, Algorithm 3], because it performed worse and the constants in the estimate for the linear convergence rate are suboptimal.

Figure 7 analyzes the differences in convergence depending on the number of inner iterations and whether the inner problem is formulated as the primal or the dual problem. In general, we found 10 inner iterations to be a good choice for the iteratively reweighted algorithms, though the optimal choice in this particular example is 5 iterations. For IRLS, the best performance is achieved by solving the *primal* inner problem, whereas for IRHuber is is advantageous to solve the *dual* problem.

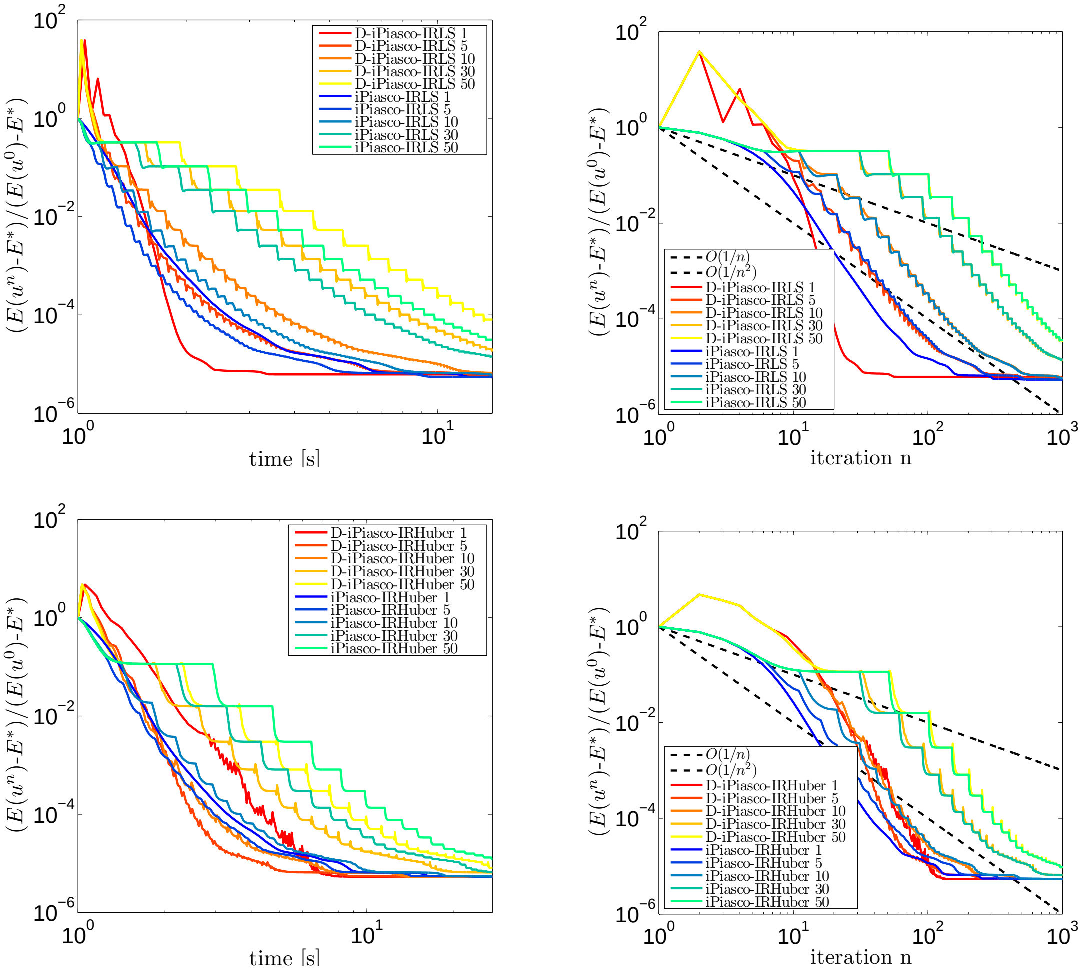


Figure 7: Convergence depending on the solver used for the inner problem (primal or dual) and the number of inner iterations (number in the figure legend). In general, 5-10 iterations is a good choice. In terms of actual computation time, for IRLS using the primal of the inner problem yields the fastest convergence; for IRHuber it is the dual of the inner problem. Both convex surrogate functions are solved with algorithms that have an (optimal) linear convergence rate.

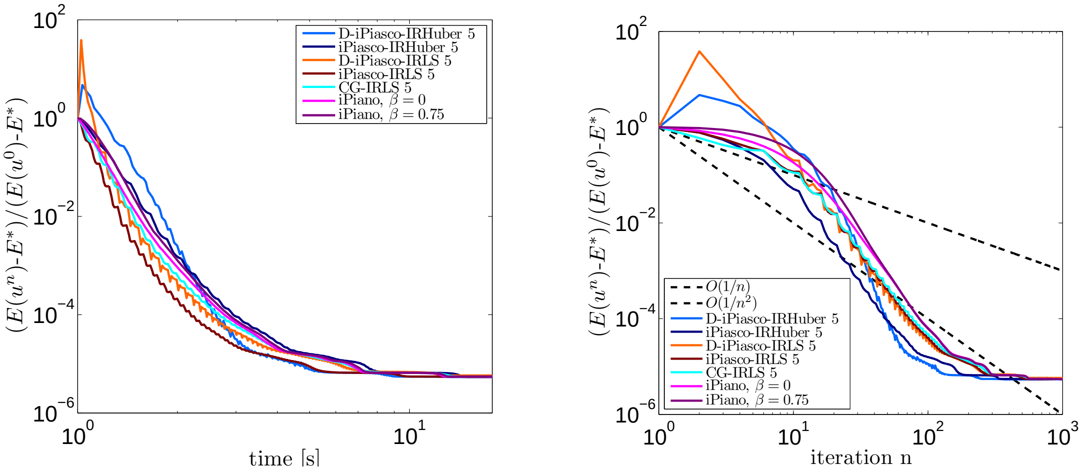


Figure 8: Comparison between IRLS, IRHuber, and iPiano. The iteratively reweighted algorithms perform best in this experiment. Regarding actual computation time IRLS is the fastest, whereas regarding iterations IRHuber shows the best convergence rate.

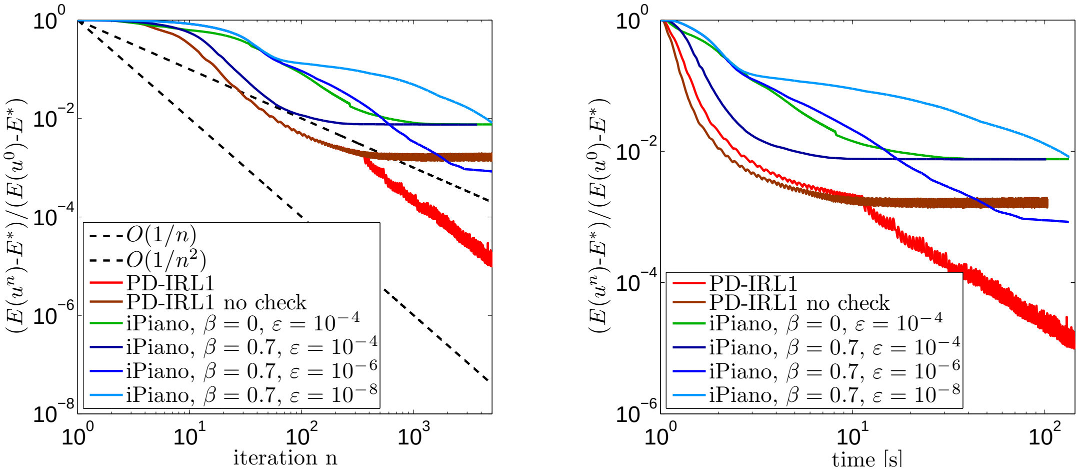


Figure 9: Comparison of the energy decrease for problem (26) between IRL1 and different parametrizations of iPiano. As our algorithm can optimize the non-regularized energy in (26) it achieves a lower energy. The version *PD-IRL1 no check* does not check the energy decrease but updates the weights every 10th iteration. It is the fastest in this experiment up to an accuracy of about $10^{-3}$, then *PD-IRL1* takes over.

Figure 8 shows the comparison of the energy decrease between IRHuber and IRLS. In terms of actual computation time IRLS performs better. This is due to the split definition of the Huber function, which additionally requires to distinguish two cases (norm less or greater than $\varepsilon$). As this extra computation cost does not matter in terms of the number of iterations, IRHuber converges the quickest in that case. Regarding both, number of iterations and computation time, the iteratively reweighted algorithms perform better than iPiano.

**Iteratively reweighted $\ell_1$ on TV-term.** As mentioned before, our iteratively reweighted $\ell_1$ algorithm is not well suited for problems that have a quadratic behavior around 0. The cases where the IRL1 algorithm becomes interesting is beyond the applicability of iPiano, namely for instance when $F_2(|u|) = \log(1 + |u|)$. iPiano can only be applied to a smoothed version of $F_2$. However, then, a different energy is minimized. We evaluate the IRL1 algorithm on the following objective:

$$\min_{u \in \mathbb{R}^{154401}} \|u - f\|_1 + \sum_i \log(1 + |Du|_i) \tag{26}$$

and use $\log(1 + |Du|_{i,\varepsilon})$ for iPiano. The input $f$ and the visual result are shown in Figure 10. The numeric comparison against iPiano with backtracking (nmiPiano in [67]) is shown in Figure 9. On one hand, reducing the $\varepsilon$ in the regularization of $|Du|_{i,\varepsilon}$ better approximates the original problem, but on the other hand, the problem is more difficult to solve for iPiano and needs many more iterations. The Lipschitz constant for iPiano depends on $\varepsilon$ and therefore directly influences the feasible step-sizes. The method *PD-IRL1 no check* finds a worse local optimum than *PD-IRL1* and *iPiano ($\beta = 0.7$, $\varepsilon = 10^{-8}$)* with a difference of about $10^{-3}$ to the "optimal" one $E^*$. It is faster in terms of actual computation time than *PD-IRL1*, because it does not have to compute the energy. *PD-IRL1* achieves a better local optimum by doing more iterations if required. IRL1 performs better than iPiano in terms of speed and, as it optimizes the original energy achieves a higher accuracy.

**Iteratively reweighted $\ell_1$ on TGV-term.** As a last numerical experiment we consider the total generalized based variation model

$$\min_{u \in \mathbb{R}^N, v \in \mathbb{R}^{2N}} \frac{\lambda}{2}\|u - f\|_2^2 + \left(\frac{\alpha_1}{\mu} \sum_i \log(1 + \mu|Du - v|_i) + \frac{\alpha_0}{\mu} \sum_i \log(1 + \mu|Dv|_i)\right). \tag{27}$$

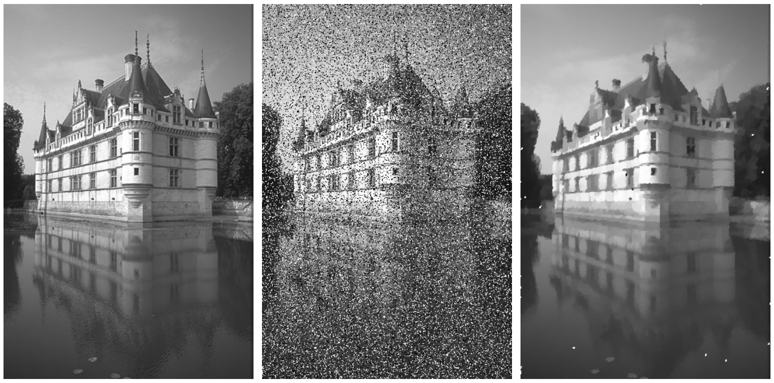


Figure 10: Visualization of the experiment (26).

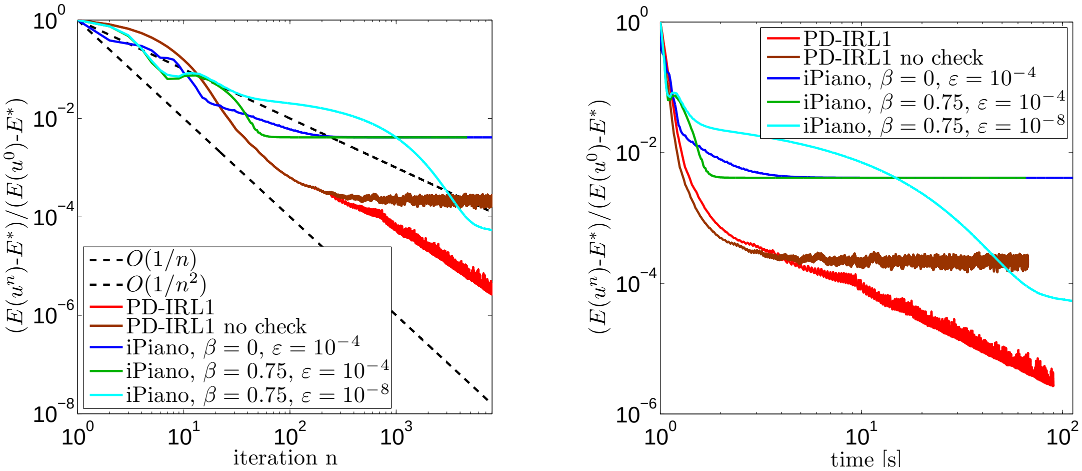
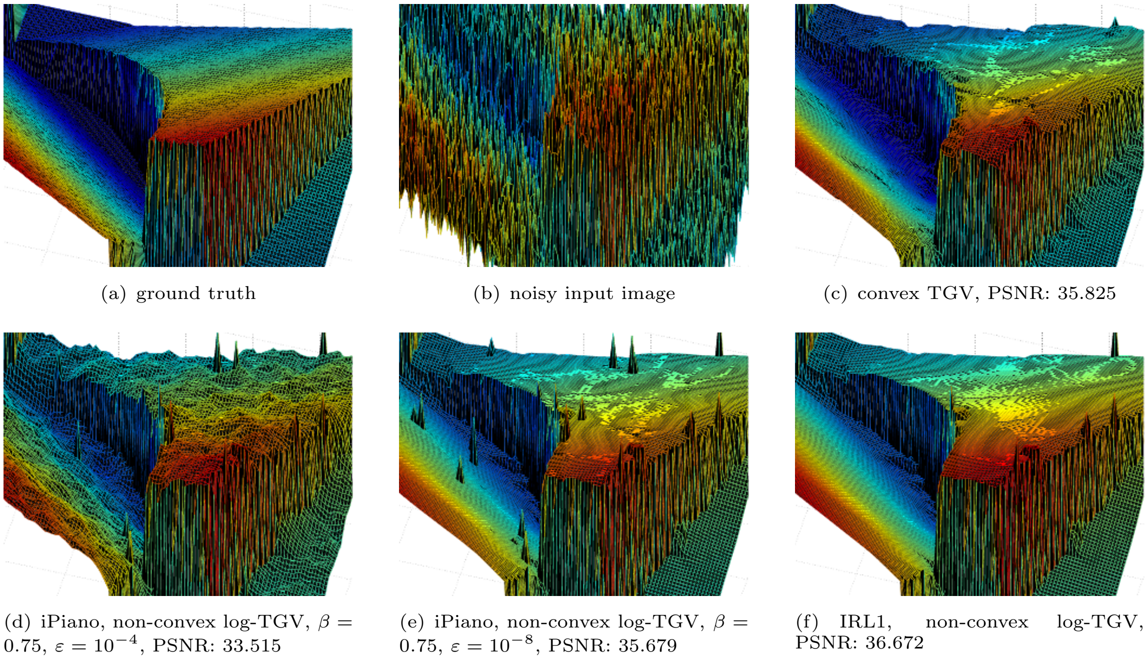


Figure 11: Comparison of the energy decrease for problem (27) between our method and different parametrizations of iPiano. As our algorithm can optimize the non-regularized energy it achieves a lower energy. Furthermore, as we solve a sequence of convex problems, we can benefit from efficient convex programming algorithm like [23]. The version *PD-IRL1 no check* does not check the energy decrease but updates the weights every 10th iteration. It is the fastest in this experiment.

The experiment is performed on an image whose 3D-mesh is shown in Figure 12(a). Its dimension is $N = 16384$. We compare our IRL1 algorithm against the iPiano algorithm (with backtracking) on an $\varepsilon$-regularized energy (nmiPiano in [67]) for the non-convex TGV model in terms of convergence. The model parameters are set to $\mu = 8$, $\lambda = 4$, $\alpha_1 = 0.5$, and $\alpha_0 = 1$. In Figure 11 the energy decrease for different methods is plotted. From the optimization viewpoint, it is well-known [17, 72] that the TGV-regularization model is a hard problem even in the convex case. In [72], the problem is efficiently solved using the primal dual algorithm [23], which is also used here for the convex surrogate problems. As for our algorithm a sequence of convex problems arises, we can benefit from efficient convex programming algorithms. Therefore, we observe a faster convergence for the IRL1 algorithm compared to iPiano. The difference becomes more and more significant the smaller the $\varepsilon$ is chosen for making the TGV differentiable in 0.

## 7.5 Total generalized variation experiment

As the TGV-regularizer is developed only recently and first used in the non-convex setting here, we perform another experiment with the energy model (27) where the focus is on accuracy. Figure 12 (c) and (f) compare the convex TGV with the non-convex TGV. For each of them, the parameters are optimized with respect to the PSNR value, which is 35.835 for the convex model and 36.672 for the non-convex model. In (c) we set $\lambda = 1$, $\alpha_1 = 0.1$, and $\alpha_0 = 1$ and in (f) we set $\mu = 8$, $\lambda = 4$, $\alpha_1 = 0.5$, and $\alpha_0 = 1$.


(a) ground truth

(b) noisy input image

(c) convex TGV, PSNR: 35.825

(d) iPiano, non-convex log-TGV, $\beta = 0.75$, $\varepsilon = 10^{-4}$, PSNR: 33.515

(e) iPiano, non-convex log-TGV, $\beta = 0.75$, $\varepsilon = 10^{-8}$, PSNR: 35.679

(f) IRL1, non-convex log-TGV, PSNR: 36.672

Figure 12: Comparison of TGV regularization for the noisy input (b) using the model (27). In (c) the result in the convex setting is shown and in (d), (e) and (f) the non-convex setting with log-norm. The comparison between (c) and (f) shows that non-convex penalizers are the better choice, and the comparison between (d), (e) and (f) shows the importance of solving the non-regularized energy model. As the result with IRL1 has fewer spikes and the energy is smaller, it found a better local optimum.

Non-convex norms in the regularization is a good choice when (1) sharp discontinuities are desired or (2) the properties of the regularizer (here the ability to reconstruct piecewise affine functions) are to be enforced.

On the other hand, the comparison between (d), (e) and (f) reveals the importance of solving a non-regularized energy model. The result (f) is nicely piecewise smooth. The problem in (d) and (e) of several small outliers does not arise in the IRL1 algorithm as the first inner subproblem is the convex TGV-model which yields already a smooth result. Then, in the next iterations discontinuities are enhanced again.

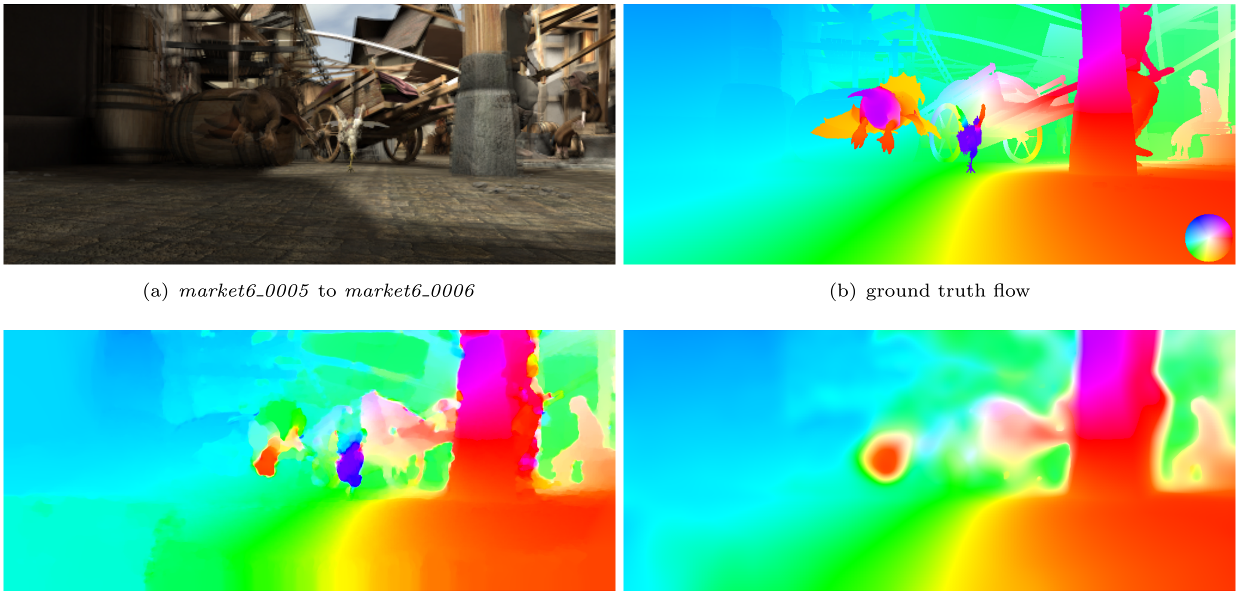
# 8 Optical flow estimation with non-convex regularizers

In this section we show application examples of non-convex energy models for the task of optical flow estimation. The same modeling principles can be transferred directly to many other vision tasks, as demonstrated in our conference paper [68], where we showed examples on denoising, deconvolution, depth map fusion, and optical flow estimation. Here we focus on a more detailed analysis of optical flow including non-convex data terms and non-convex regularizers and their effects.

Optical flow describes dense correspondences between a pair of images $f(x,t)$ and $f(x,t+1)$. Modeling of variational optical flow usually consists of a regularization term and a data term. The first one models the smoothness of the flow field. The data term measures the difference between the motion compensated second frame and the image of the first frame. A simple example is the difference of gray values (brightness constancy assumption) $\|f(x+u(x),t+1) - f(x,t)\|_2^2$, where $u = (u^1,u^2)^\top$ is the sought optical flow field. Since the unknown flow field is a variable of the generally non-convex image, the data term is non-convex independent of the properties of the penalty function. In practice, this kind of non-convexity is dealt with by a Gauss-Newton scheme in combination with a continuation method [18][1].

Previous works on variational optical flow estimation always employ a convex penalty function for the data term and the regularizer. The results of the following experiments shall indicate that one can benefit from

[1] It cannot be approached well with our algorithm.



(a) *market6_0005* to *market6_0006*

(b) ground truth flow

(c) LDOF [19], EP: 13.18

(d) over-smoothed flow field, EP: 6.59

Figure 13: Considering only the quantitative result shows that the method from (d) is better than (c). However, the over-smoothed result in (d) is practically useless compared to (c).

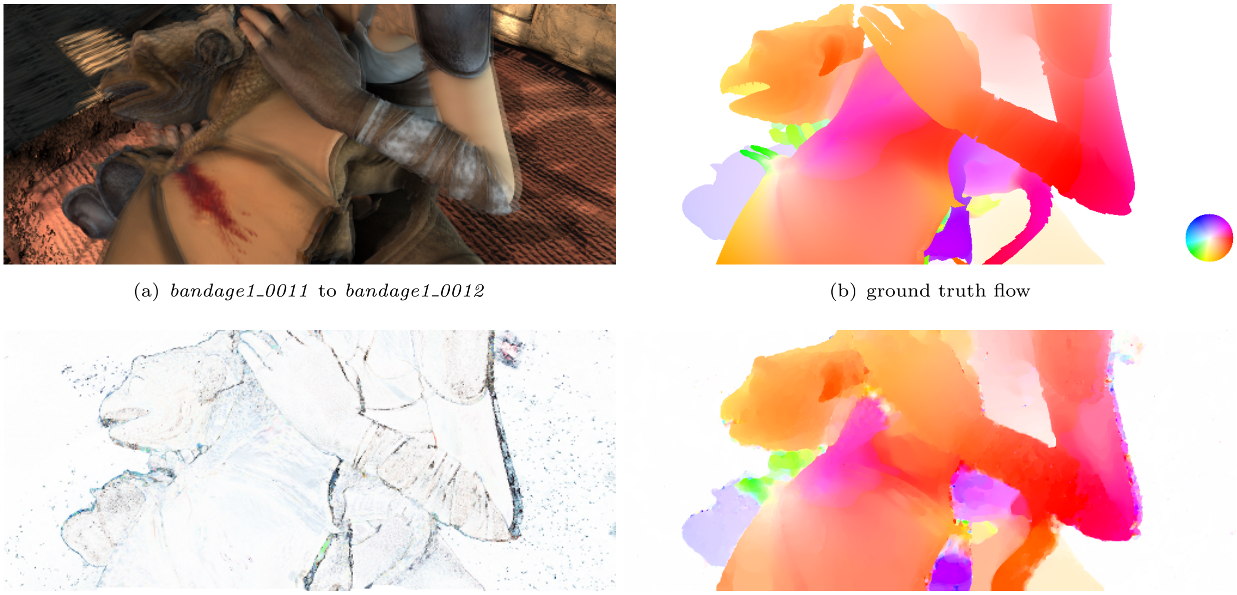
rather using non-convex penalties. Details such as parameter optimization, the optimal usage in conjunction with a continuation method, etc., need further investigation to compete with heavily tuned methods on standard benchmark datasets. Such optimization is not in the scope of this paper. In the following, we present three ways to apply and use the non-convex penalties with the algorithm proposed in this paper: a non-convex penalty on the brightness constancy assumption, a non-convex regularizer, and a non-convex penalty for integrating point correspondences into variational methods.

The experiments are performed on image pairs (clean version) from the Sintel benchmark [20][2]. The standard quality measure of the Sintel benchmark is the average endpoint error. It is well-known that average errors emphasize global properties of the flow fields while details, such as sharp discontinuities, are under-represented. Figure 13 shows an example. This fact is disadvantageous for non-convex regularization penalties, which are particularly good for obtaining sharp discontinuities. For this reason, we present mainly qualitative results but also report the endpoint errors.

## 8.1 Non-convex data term: robust optical flow

Outliers in the data term, mostly caused by occlusion, are a major issue in optical flow estimation. There are two aspects of this problem: detection of occluded points and interpolation of the flow field at these points. Non-convex penalty functions allow for a straightforward approach to implicitly deal particularly with the first aspect. The basic assumption for estimating the optical flow is the brightness constancy (or color constancy) assumption. As it is typically not satisfied in occlusion areas, the penalty on the brightness constancy should be reduced there. This is naturally achieved by non-quadratic penalties, which implicitly weigh down the influence of points that contradict the constancy assumption. With convex penalty functions, however, the effect is often not strong enough. With non-convex penalty functions, the influence of outliers can be reduced much more. In the limit, this approaches the algorithmic two-step treatment, where outliers are first detected explicitly and then removed completely from the estimation process.

[2] The Sintel benchmark provides ground truth optical flow fields for a realistically rendered video. It provides three different stages of this rendering process: *albedo*, *clean*, and *final*.



(a) *bandage1_0011* to *bandage1_0012*

(b) ground truth flow

(c) final data term weighting

(d) flow estimated using (28)

Figure 14: The weighting mask is part of the optimization using our algorithm due to using a robust, non-convex norm. It is not introduced separately. The algorithm automatically weights down the brightness constancy assumption where it is violated (darker areas in (c)). Each color channel is weighted individually and the corresponding color represents the weighting.

As in occlusion areas the measured data is invalid, the optical flow field in these areas must be inferred by prior knowledge like smoothness of the flow field. In a variational formulation, which seeks for a global agreement of all constraints, the reduction of the penalty on the brightness constancy assumption automatically assigns more importance to the regularization term in these areas. It is an open question what is the best regularizer for this job. We consider regularization with total variation, which is easy to use and preserves discontinuities. It could be easily replaced by some other, more complex prior.

We consider the following energy model:

$$\min_u \sum_i |\nabla u|_i + \frac{\lambda}{\mu} \sum_{i,k} \log(1 + \mu |\rho^k(u)|_i) \,, \tag{28}$$

where $\rho^k(u) = f_t^k + (\nabla f^k)^\top (u - u_0)$ implements the linearized brightness constancy assumption for the color channel $k \in \{1, 2, 3\}$, and $\lambda = 15$, $\mu = 5$.

We minimize the energy with the iteratively reweighted $\ell_1$ algorithm. The algorithm generates an automatic weighting for the data term. The weights are small where the brightness constancy assumption does not hold, i.e., for outliers. The weighting is directly inferred from the cost function. The approach in [7] models occlusions explicitly, but in the end it comes down to a similar weighting. Figure 14 shows an example. As expected, the weighting in Figure 14(c) shows a reduction of the penalty particularly in occlusion areas.

Despite the quite good detection of the occlusion region due to the non-convex penalty, the optical flow of the arms still leaks into the background. This is due to the weak smoothness prior, which does not take the direction of the occlusion into account. Future work on smoothness priors may exploit the detected occlusion regions more effectively.

## 8.2 Non-convex TGV regularized optical flow

Total variation is the most popular regularizer for the optical flow field. However, it penalizes also flow fields that describe rotation and scaling motion. Total generalized variation deals with this problem, as affine motion can be described without penalty. Therefore, we consider the variational model given by

$$\min_{u,v} \lambda Q(f,u) + \alpha_1 F_{2,1}(|\nabla u - v|) + \alpha_0 F_{2,0}(|\nabla v|), \tag{29}$$

where $Q(f,u)$ is the quadratic fitting term from [83] using normalized cross correlation, and $F_{2,1}$, $F_{2,0}$ are (possibly different) non-convex penalty functions. The model described in (29) can be used to enforce the TGV-properties by using non-convex norms. This yields highly desirable sharp motion discontinuities as can be seen in the bottom row of Figure 15. The penalty on $|\nabla v|$ can be seen as a penalty on kinks in the flow field. Reducing the cost of sharp kinks by a non-convex penalty often leads to the fact that sharp discontinuities in the flow field are replaced by a linear transition with two sharp kinks. Therefore, we set $F_{2,0} = \mathrm{id}$.

### 8.3 Non-convex integration of point correspondences

Current state-of-the-art methods [85, 19, 82] usually incorporate a sparse or semi-dense feature matching into the optimization procedure. In LDOF [19], the deviation of the estimated flow field from these feature matches is penalized. The penalty is based on the $\ell_1$-norm. Non-convex norms are more robust and can deal with more erroneous feature matches in a certain local area than the $\ell_1$-norm. In the following model, we propose to use a non-convex penalty function for the deviation from the initial feature matches:

$$\min_{u,v} \lambda \|Q(f,u)\|_1 + \frac{\beta}{p}\|u - u_{\mathrm{FM}}\|_{p,\varepsilon} + \alpha_1 \|\nabla u - v\|_1 + \alpha_0 \|\nabla v\|_1, \tag{30}$$

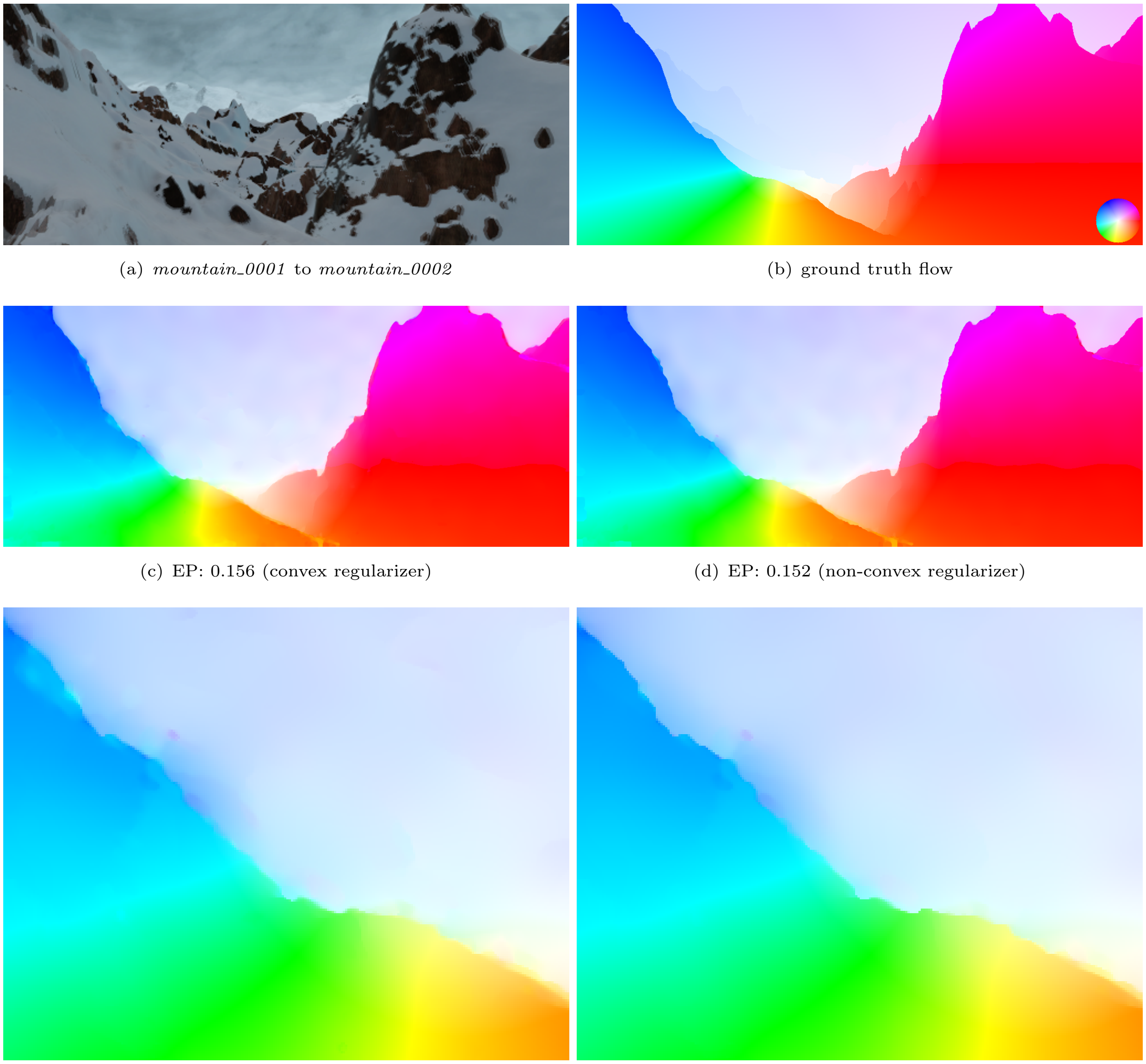
where $Q(f,u)$ is the quadratic fitting term from [83] using normalized cross correlation, $u_{\mathrm{FM}}$ are sparse feature correspondences estimated like in [19], and $p \leq 1$ determines the non-convexity of the feature matching penalizer. Figure 16 compares the convex vs. the non-convex penalty term. In this formulation, we evaluate a convex energy ($cFMcTGV$) with $p = 1, \varepsilon = 0$, $\lambda = 1$ and $\beta = 2$ versus a non-convex energy ($ncFMcTGV$) with $p = 0.5, \varepsilon = 0.001$, $\lambda = 0.8$ and $\beta = 3$. Both settings use $\alpha_1 = 0.2$ and $\alpha_0 = 1$. All parameters were optimized for two challenging image pairs of the Sintel optical flow benchmark [20]. The two image pairs are chosen complementary in the way feature matches should be used. For *market6_0005*, many feature matches (bottom of the image) should be considered as outliers, whereas for *cave2_0015* the few correct feature matches on the dragon must be used to capture the large motion. Our results show that feature matching driven optical flow methods can benefit a lot from non-convex penalty functions.

## 9 Conclusion

In this paper, we proposed a general algorithm for a certain class of non-smooth non-convex optimization problems: the sum of a convex function and a composition of a coordinate-wise convex function with a non-convex function. It is a majorization-minimization (MM) algorithm that contains the iteratively reweighted least squares and the iteratively reweighted $\ell_1$ algorithm as special cases. Moreover, we introduced another two special instances of the algorithm with favorable properties.

As the proposed algorithm is a MM algorithm, it yields non-increasing function values, which together with coercivity implies the existence of a converging subsequence for proper functions. In the second part of our convergence analysis, convergence for the whole sequence of iterates was proved under additional mild assumptions such as Lipschitz continuity of the gradient of the non-convex part of the objective function. Assuming that the objective has the Kurdyka-Łojasiewicz property at accumulation points, we benefit from a recently proved convergence result for abstract descent methods. A careful analysis of our situation allowed us to verify the conditions under which this abstract theorem holds.

The second part of the paper is devoted to the numerical analysis. It was shown that starting with different initializations the algorithm mostly ends up in the same local optimum and that the proposed algorithm converges quickly. Particularly for non-smooth, non-convex problems other methods are only applicable after regularization, which usually makes the convergence dependent on the regularization parameter. There is



(a) *mountain_0001* to *mountain_0002*

(b) ground truth flow

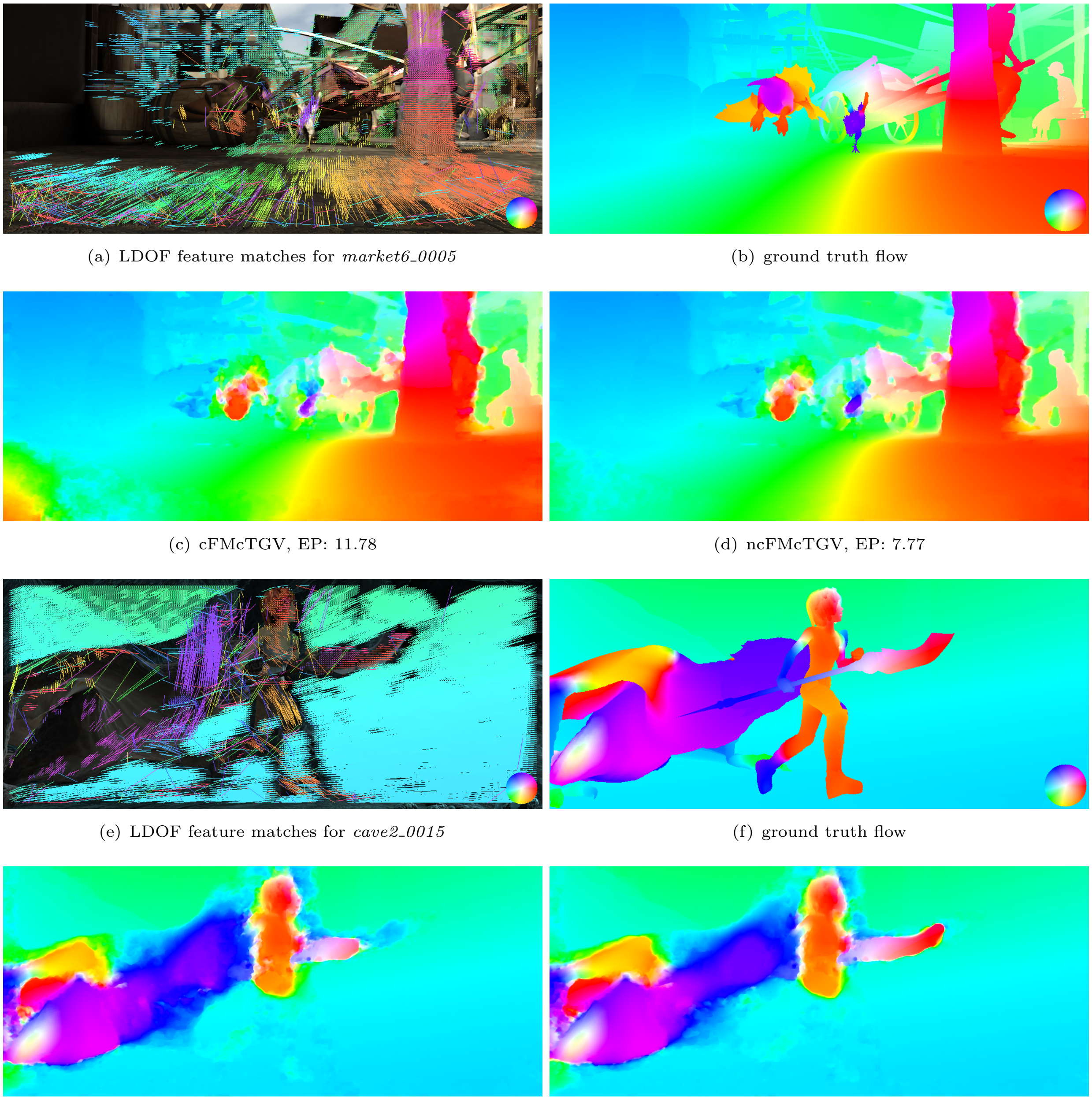
(c) EP: 0.156 (convex regularizer)

(d) EP: 0.152 (non-convex regularizer)

(e) zoom into (c) (convex regularizer)

(f) zoom into (d) (non-convex regularizer)

Figure 15: (c) is obtained with model (29) and $\lambda = 0.3$, $\alpha_1 = 0.1$, $\alpha_0 = 1.0$, $F_{2,1} = \| \cdot \|_2$, $F_{2,0} = \| \cdot \|_2$, and (d) using the parameters $\lambda = 0.25$, $\alpha_1 = 0.1$, $\alpha_2 = 1.0$, $F_{2,1}(|x|) = 2 \log(1 + \frac{1}{2}|x|)$, $F_{2,0} = \| \cdot \|_2$. The result in (d) and its zoom (f) show that using non-convex penalizers are beneficial and yield sharp discontinuities.



(a) LDOF feature matches for *market6_0005*

(b) ground truth flow

(c) cFMcTGV, EP: 11.78

(d) ncFMcTGV, EP: 7.77

(e) LDOF feature matches for *cave2_0015*

(f) ground truth flow

(g) cFMcTGV, EP: 9.20

(h) ncFMcTGV, EP: 8.79

Figure 16: Result for two image pairs from the Sintel optical flow benchmark using the model (30). The two image pairs are chosen because they require a complementary usage of the feature matches, in (a) many matches at the bottom should be considered as outliers, whereas in (e) the few matches on the dragon are important. With the algorithm proposed in this paper the usage of a robust penalty for the feature matching term is possible. Such a robust penalty is used for *ncFMcTGV*. In *cFMcTGV* a $\ell_1$-penalty is used. The parameters for both methods have been optimized. *ncFMcTGV* can deal with the two complementary requirements of the feature correspondences much better than *cFMcTGV*.

always a trade-off between approximation accuracy (with the regularization parameter) and speed, which is not the case for our algorithm. This is also demonstrated on a computer vision example with non-convex total generalized variation (TGV) regularization (special penalization of first and higher order derivatives). To the best of our knowledge, we are the first who consider and solve such non-convex variants of TGV regularization.

Finally, in order to prove the practical impact of our algorithm, we applied it to several situations with non-convex terms in optical flow estimation and observed a consistent improvement compared to corresponding convex penalty terms.

# 10 Acknowledgements

Thomas Pock acknowledges support from the Austrian science fund (FWF) under the projects No. I1148 and No. Y729. Peter Ochs, Alexey Dosovitskiy and Thomas Brox acknowledge funding by the German Research Foundation (DFG grant BR 3815/5-1) and the ERC Starting Grant VideoLearn. Moreover, the authors thank the anonymous reviewers for their valuable and constructive comments.

# 11 Appendix

## 11.1 Mathematical preliminaries

We review here some definitions and results from [76].

**Definition 1** (domain)**.** *The* domain *of a function* $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ *is the set* $\operatorname{dom} f := \{x \in \mathbb{R}^n : f(x) < \infty\}$*. The* domain *of a point-to-set mapping* $F\colon \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ *is the set* $\operatorname{dom} F := \{x \in \mathbb{R}^n : F(x) \neq \emptyset\}$*.*

**Definition 2** (normal vectors, normal cone)**.** *Let* $C \subset \mathbb{R}^n$ *and* $\bar{x} \in C$*. A vector* $v$ *is a* regular normal *vector to* $C$ *at* $\bar{x}$*, written* $v \in \widehat{N}_C(\bar{x})$*, if*

$$\limsup_{\substack{x \xrightarrow[C]{} \bar{x} \\ x \neq \bar{x}}} \frac{\langle v, x - \bar{x}\rangle}{\|x - \bar{x}\|_2} \leq 0\,,$$

*where* $x \xrightarrow[C]{} \bar{x}$ *means all sequences* $(x^\nu)_{\nu \in \mathbb{N}}$ *converging to* $\bar{x}$ *with* $x^\nu \in C$ *for all* $\nu \in \mathbb{N}$*. It is a* (general) normal *vector to* $C$ *at* $\bar{x}$*, written* $v \in N_C(\bar{x})$*, if there are sequences* $x^\nu \xrightarrow[C]{} \bar{x}$*,* $v^\nu \to v$ *with* $v^\nu \in \widehat{N}_C(x^\nu)$*.*

**Definition 3** (Clarke regularity)**.** *A function* $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ *is called* (Clarke) regular *at* $\bar{x}$ *if* $f(\bar{x})$ *is finite and the epigraph* $\operatorname{epi} f := \{(x,t) | x \in \operatorname{dom} f, t \geq f(x)\}$ *is* Clarke regular *at* $(\bar{x}, f(\bar{x}))$ *as a subset of* $\mathbb{R}^n \times \mathbb{R}$*, i.e.,* $\operatorname{epi} f$ *is locally closed and it holds* $N_{\operatorname{epi} f}(\bar{x}) = \widehat{N}_{\operatorname{epi} f}(\bar{x})$*.*

We need the following generalization of the subgradient of convex functions [76, Def. 8.3].

**Definition 4** (limiting-subgradient, regular subgradient, horizon subgradient)**.** *For a function* $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ *and a point* $\bar{x} \in \operatorname{dom} f$

1. *the* subgradient *(or* limiting-subgradient*) is defined by*

$$\partial f(\bar{x}) = \{v \in \mathbb{R}^n | \, \exists x^\nu \to \bar{x},\, f(x^\nu) \to f(\bar{x}),\, v^\nu \to v,\, v^\nu \in \widehat{\partial} f(x^\nu)\}\,, \tag{31}$$

*which makes use of the* regular *subgradient defined by*

$$\widehat{\partial} f(\bar{x}) = \{v \in \mathbb{R}^n | \, \liminf_{\substack{x \to \bar{x} \\ x \neq \bar{x}}} \tfrac{1}{\|x - \bar{x}\|_2} \left(f(x) - f(\bar{x}) - \langle x - \bar{x}, v\rangle\right) \geq 0\}\,.$$

*2. The* horizon subgradient *is defined by*

$$\partial^\infty f(\bar{x}) = \{v \in \mathbb{R}^n |\, \exists x^\nu \to \bar{x},\, f(x^\nu) \to f(\bar{x}),\, \exists \lambda^\nu \searrow 0 : \lambda^\nu v^\nu \to v,\, v^\nu \in \widehat{\partial} f(x^\nu)\}\,.$$

This definition allows the optimality of a point $\hat{x} \in X$ to be characterized by $0 \in \partial F(\hat{x})$. Such a $\hat{x}$ is called a *stationary point.*

The following corollary relates this definition to the notion of subgradient and regular subgradient ([76, Cor. 8.11])

**Corollary 1.** *For a function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite and $\partial f(\bar{x}) \neq \emptyset$, one has $f$ regular at $\bar{x}$ implies that $f$ is locally lsc at $\bar{x}$ with*

$$\partial f(\bar{x}) = \widehat{\partial} f(\bar{x})\,.$$

An important concept for many results in variational analysis is Lipschitz continuity. We formulate it as defined in [76, Def. 9.1].

**Definition 5** (Lipschitz continuity)**.** *Let $F\colon D \to \mathbb{R}^m$ be a single-valued mapping defined on a set $D \subset \mathbb{R}^n$ and let $X \subset D$. Then, we define $F$ is* Lipschitz continuous *on $X$ if there exists $L \in [0, \infty)$ with*

$$\|F(x) - F(y)\|_2 \leq L\|x - y\|_2, \quad \textit{for all } x, y \in X\,.$$

*Then $L$ is called the* Lipschitz constant *for $F$ on $X$.*

For the convergence analysis of our algorithm a certain class of functions, namely *functions with Lipschitz continuous gradient* will be of importance. A continuously differentiable function $f\colon X \to \mathbb{R}$ with $X \subset \mathbb{R}^n$ open and with Lipschitz continuous gradient $L > 0$ satisfies by definition

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2 \quad \forall x, y \in X\,.$$

Another property for such functions is given by the following lemma.

**Lemma 2.** *Let $f\colon X \to \mathbb{R}$ have Lipschitz continuous gradient with $L > 0$ and let $X \subset \mathbb{R}^n$ open. Then for any $y \in X$ it holds that*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|_2^2\,, \quad \textit{for all } x \in X\,. \tag{32}$$

*Proof.* See [60]. □

**Proposition 7.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ have Lipschitz continuous gradient, then $f$ is a regular function.*

*Proof.* See [76, Thm. 9.18]. □

Where Lipschitz continuity of the gradient of a function is important to find quadratic majorizers of the function, *strong convexity* provides a way to access quadratic minorizers for convex functions. This property can be defined as follows.

**Definition 6** (strongly convex function)**.** *A proper function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ is called* strongly convex*, if there exists $\mu > 0$ such that $f(x) - \mu/2\|x\|_2^2$ is a convex function. The parameter $\mu$ is denoted the* convexity parameter*.*

**Lemma 3.** *Let $f\colon X \to \overline{\mathbb{R}}$, $X \subset \mathbb{R}^n$, be a proper, strongly convex function with convexity parameter $\mu > 0$. Then for any $y \in \operatorname{dom} \partial f$ it holds*

$$f(x) \geq f(y) + \langle \xi, x - y \rangle + \frac{\mu}{2}\|x - y\|_2^2\,, \quad \textit{for all } \xi \in \partial f(y) \textit{ and } x \in X\,. \tag{33}$$

*Proof.* The statement is a simple consequence of Definition 6. □

The following proposition combines the definition of Lipschitz continuity with [76, Thm. 9.13].

**Proposition 8** (relation horizon and limiting subgradient)**.** *Let $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ be a locally lower semi-continuous (lsc) function with finite value at $\bar{x}$. Then the following statements are equivalent:*

1. *$f$ is locally Lipschitz continuous at $\bar{x}$,*
2. *$\partial^\infty f(\bar{x}) = \{0\}$,*
3. *$\partial f : x \mapsto \partial f(x)$ is locally bounded at $\bar{x}$,*
4. *$\widehat{\partial} f : x \mapsto \widehat{\partial} f(x)$ is locally bounded at $\bar{x}$.*

The next proposition relates the subgradient of a sum of functions to the sum of the subgradients of functions (see [76, Cor. 10.9])

**Proposition 9** (addition of functions)**.** *Suppose $f = f_1 + \ldots + f_m$ for proper, lsc functions $f_i\colon \mathbb{R}^n \to \overline{\mathbb{R}}$, and let $\bar{x} \in \operatorname{dom} f$. Then*

$$\widehat{\partial} f(\bar{x}) \supset \widehat{\partial} f_1(\bar{x}) + \ldots + \widehat{\partial} f_m(\bar{x})\,.$$

*If the only combination of vectors $v_i \in \partial^\infty f_i(\bar{x})$ with $v_1 + \ldots + v_m = 0$ is $v_1 = \ldots = v_m = 0$, one also has that*

$$\partial f(\bar{x}) \subset \partial f_1(\bar{x}) + \ldots + \partial f_m(\bar{x})\,.$$

*If also each $f_i$ is Clarke regular at $\bar{x}$, then $f$ is Clarke regular at $\bar{x}$ and*

$$\partial f(\bar{x}) = \partial f_1(\bar{x}) + \ldots + \partial f_m(\bar{x})\,.$$

In our analysis, we will need an extended version of the chain rule that applies to compositions of non-smooth functions as in [76, Thm. 10.49].

**Proposition 10** (extended chain rule)**.** *Let $f(x) = F(G(x))$ for a proper, lsc function $F\colon \mathbb{R}^m \to \overline{\mathbb{R}}$ and a locally Lipschitz continuous vector-valued function $G\colon X \to \mathbb{R}^m$, $X \subset \mathbb{R}^n$. Then, for $\bar{x} \in X$ it holds*

$$\widehat{\partial} f(\bar{x}) \supset \widehat{D}^* G(\bar{x})[\widehat{\partial} F(G(\bar{x}))] = \bigcup \{\widehat{\partial} \langle y, G \rangle (\bar{x}) |\, y \in \widehat{\partial} F(G(\bar{x}))\}\,.$$

*If the only vector $y \in \partial^\infty F(G(\bar{x}))$ with $0 \in \partial \langle y, G \rangle (\bar{x})$ is $y = 0$, one also has*

$$\partial f(\bar{x}) \subset D^* G(\bar{x})[\partial F(G(\bar{x}))] = \bigcup \{\partial \langle y, G \rangle (\bar{x}) |\, y \in \partial F(G(\bar{x}))\}\,,$$

*If in addition $F$ is regular at $G(\bar{x})$ and $\langle y, G \rangle$ is regular at $\bar{x}$ for each $y \in \partial F(G(\bar{x}))$, then $f$ is regular at $\bar{x}$ and $\partial f(\bar{x}) = D^* G(\bar{x})[\partial F(G(\bar{x}))]$.*

Finally, the convergence analysis will only be valid for functions satisfying the so called Kurdyka-Łojasiewicz (KL) property. From a practical point of view, this is not a restriction as almost all functions of practical relevance in computer vision or machine learning have this property. The KL property was originally introduced in [56] for smooth functions and generalized in [52] for functions with o-minimal structure. Now, it is known that real analytic, semi-algebraic or more generally globally subanalytic functions have the KL property [14], and even more general lower semi-continuous functions that are definable in an o-minimal structure [15]. Recently, it became widely used for proofs of convergence for non-smooth non-convex functions. Before we give a proper definition of the KL property, we consider the definition of semi-algebraic functions, which provide a rich class of KL functions.

**Definition 7** (semi-algebraic sets and functions (from [6]))**.**

1. *A subset $S$ of $\mathbb{R}^n$ is a* real semi-algebraic set *if there exists a finite number of real polynomial functions $f_{i,j}, g_{i,j} \colon \mathbb{R}^n \to \mathbb{R}$, $1 \le i \le q$, $1 \le j \le p$ such that*

$$S = \bigcup_{j=1}^{p} \bigcap_{i=1}^{q} \{x \in \mathbb{R}^n : f_{i,j}(x) = 0, g_{i,j}(x) < 0\}\,.$$

2. *A function $f \colon \mathbb{R}^n \to \overline{\mathbb{R}}$ is called a* semi-algebraic *function if its graph $\{(x, \lambda) \in \mathbb{R}^{n+1} : f(x) = \lambda\}$ is a semi-algebraic subset of $\mathbb{R}^{n+1}$.*

Simple examples of semi-algebraic functions are polynomials, $\|\cdot\|_2$, or indicator functions of semi-algebraic sets. In order to generate some more examples, usually the behavior under combination with certain operations is considered. It can be shown (see e.g. [13]):

- Semi-algebraic sets are stable under finite unions, finite intersections, complementation, and Cartesian products.
- The image and preimage of a semi-algebraic set under a semi-algebraic function is semi-algebraic.
- Finite sums and products of semi-algebraic functions are semi-algebraic.
- Compositions of semi-algebraic functions are semi-algebraic.

Although we do not define o-minimal structures, it is worth mentioning that semi-algebraic functions are definable in an o-minimal structure. In some definitions of o-minimal structures, the class of semi-algebraic sets is even part of the definition. It is usually considered as the smallest o-minimal structure. Actually, o-minimal structures are an axiomatic collection of the favorable properties of semi-algebraic functions. They are stable under the same operations as those mentioned above. o-minimal structures that properly contain the semi-algebraic sets are globally subanalytic sets [40]. There is even a larger o-minimal structure, namely the log–exp structure, which comprises the globally subanalytic structure and the graph of the exponential function [32, 84]. As mentioned above, functions that are definable in such an o-minimal structure have the KL property [15, Thm. 14].

Now, as we have some examples, we consider the formulation of the Kurdyka-Łojasiewicz property as in [6].

**Definition 8** (Kurdyka-Łojasiewicz property)**.** *Let $f \colon \mathbb{R}^n \to \overline{\mathbb{R}}$ be an extended real valued function and let $\hat{x} \in \operatorname{dom} \partial f$. If there exists $\eta \in (0, \infty]$, a neighborhood $U$ of $\hat{x}$ and a continuous concave function $\varphi \colon [0, \eta) \to \mathbb{R}_+$ such that $\varphi(0) = 0$, $\varphi \in C^1((0, \eta))$, for all $s \in (0, \eta)$ it is $\varphi'(s) > 0$, and for all $x \in U \cap \{x \in \mathbb{R}^n : f(\hat{x}) < f(x) < f(\hat{x}) + \eta\}$ holds the Kurdyka-Łojasiewicz inequality*

$$\varphi'(f(x) - f(\hat{x}))\operatorname{dist}(0, \partial f(x)) \ge 1\,, \tag{34}$$

*then the function has the* Kurdyka-Łojasiewicz property *at $\hat{x}$.*

*If, additionally, the function is lower semi-continuous and the property holds for each point in* $\operatorname{dom} \partial f$*, then $f$ is called a* Kurdyka-Łojasiewicz function*.*

It is easy to see that the Kurdyka-Łojasiewicz property is satisfied for all non-stationary points (see eg. [5]). For the purpose of intuition, it should be mentioned, that for smooth functions (with $f(\hat{x}) = 0$) (34) is equivalent to $\|\nabla(\varphi \circ f)(x)\|_2 \ge 1$. This means, that after reparametrization via $\varphi$ the gradient $\nabla f$ may be bounded away from 0.

[6] proves convergence of a sequence $(x^k)_{k \in \mathbb{N}}$ under the following assumptions. This convergence theorem is central to our analysis, as we verify the sequence generated by Method 1, which is the most general one, to obey these properties. We recap this result of [6] now.

Let $f \colon \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper lower semi-continuous function and let $a$ and $b$ be fixed positive constants. We consider the following properties:

(H1) *(Sufficient decrease condition).* For each $k \in \mathbb{N}$,

$$f(x^{k+1}) + a\|x^{k+1} - x^k\|_2^2 \leq f(x^k)\,;$$

(H2) *(Relative error condition).* For each $k \in \mathbb{N}$, there exists $w^{k+1} \in \partial f(x^{k+1})$ such that

$$\|w^{k+1}\|_2 \leq b\|x^{k+1} - x^k\|_2\,;$$

(H3) *(Continuity condition).* There exits a subsequence $(x^{k_j})_{j\in\mathbb{N}}$ and $\tilde{x}$ such that

$$x^{k_j} \to \tilde{x} \text{ and } f(x^{k_j}) \to f(\tilde{x}), \quad \text{as } j \to \infty\,.$$

**Theorem 3** (Convergence of descent methods ([6, Theorem 2.9]))**.** *Let $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper lower semi-continuous function. Consider a sequence $(x^k)_{k\in\mathbb{N}}$ that satisfies H1, H2, and H3.*

*If $f$ has the Kurdyka-Łojasiewicz property at the cluster point $\tilde{x}$ specified in H3 then the sequence $(x^k)_{k\in\mathbb{N}}$ converges to $\bar{x} = \tilde{x}$ as $k$ goes to infinity, and $\bar{x}$ is a critical point of $f$.*

*Moreover the sequence $(x^k)_{k\in\mathbb{N}}$ has a finite length, i.e.,*

$$\sum_{k=0}^{\infty} \|x^{k+1} - x^k\|_2 \leq \infty\,.$$

# References

[1] M. Allain, J. Idier, and Y. Goussard. On global and local convergence of half-quadratic algorithms. *IEEE Transactions on Image Processing*, 15(5):1130–1142, May 2006.

[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] M. Artina, M. Fornasier, and F. Solombrino. Linearly constrained nonsmooth and nonconvex minimization. *SIAM Journal on Optimization*, 23(3):1904–1937, 2013.

[4] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1):5–16, June 2008.

[5] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, May 2010.

[6] H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[7] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3):322–338, 2012.

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

[9] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal-recovery problems. In *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2009.

[10] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, Sept. 1999.

[11] 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.

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

[13] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Springer, 1998.

[14] J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, Dec. 2006.

[15] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

[16] J. Bolte, A. Daniilidis, A. Ley, and L. Mazet. Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362:3319–3363, 2010.

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

[18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 2004.

[19] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:500–513, 2011.

[20] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conference on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[21] E. J. Candes, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted $\ell_1$ minimization. *Journal of Fourier Analysis and Applications,*, 2008.

[22] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Applied Mathematics*, 5(4):1113–1158, 2012.

[23] 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.

[24] X. Chen and W. Zhou. Convergence of the reweighted $\ell_1$ minimization algorithm for $\ell_2$-$\ell_p$ minimization. *Computational Optimization and Applications*, pages 1–15, 2013.

[25] E. Chouzenoux, J. Idier, and S. Moussaoui. A majorize minimize strategy for subspace optimization applied to image restoration. *IEEE Transactions on Image Processing*, 20(6):1517–1528, June 2011.

[26] E. Chouzenoux, A. Jezierska, J. Pesquet, and H. Talbot. A majorize-minimize subspace approach for $\ell_2$-$\ell_0$ image regularization. *SIAM Journal on Imaging Sciences*, 6(1):563–591, Jan. 2013.

[27] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, Nov. 2013.

[28] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[29] P. L. Combettes and V. R. Wajs. Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[30] D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Advances in Markov Random Fields for Vision and Image Processing.* MIT Press, 2011.

[31] 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.

[32] L. V. den Dries. *Tame Topology and ø-minimal Structures.* 150 184. Cambridge University Press, 1998.

[33] J. Douglas and J. E. Gunn. A general formulation of alternating direction methods. Part I. Parabolic and hyperbolic problems. *Numerische Mathematik*, 6:428–453, 1964.

[34] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3):293–318, June 1992.

[35] J. Fessler and A. Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on Signal Processing*, 42(10):2664–2677, Oct. 1994.

[36] J. Fessler and A. Hero. Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms. *IEEE Transactions on Image Processing*, 4(10):1417–1429, Oct. 1995.

[37] A. Florescu, E. Chouzenoux, J.-C. Pesquet, P. Ciuciu, and S. Ciochina. A majorize-minimize memory gradient method for complex-valued inverse problems. *Signal Processing*, 103:285–295, Oct. 2014.

[38] M. Fornasier and R. Ward. Iterative thresholding meets free-discontinuity problems. *Foundations of Computational Mathematics*, 10(5):527–567, 2010.

[39] M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8):989–1000, 1981.

[40] A. Gabrielov. Complements of subanalytic sets and existential formulas for analytic functions. *Inventiones mathematicae*, 125(1):1–12, 1996.

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

[42] 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.

[43] B. Goldluecke, E. Strekalovskiy, and D. Cremers. Tight convex relaxations for vector-valued labeling. *SIAM Journal on Imaging Sciences*, 6(3):1626–1664, 2013.

[44] A. A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70:709–710, 1964.

[45] B. He and X. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. *SIAM Journal on Applied Mathematics*, 5(1):119–149, Jan. 2012.

[46] M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[47] R. Horst and N. V. Thoai. Dc programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[48] J. Huang and D. Mumford. Statistics of natural images and models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 541–547, Fort Collins, CO, USA, 1999.

[49] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Journal of the American Statistical Association*, 58(1):30–37, 2004.

[50] J. Idier. Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *IEEE Transactions on Image Processing*, 10(7):1001–1009, July 2001.

[51] M. Jacobson and J. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Transactions on Signal Processing*, 16(10):2411–2422, Oct. 2007.

[52] K. Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 48(3):769–783, 1998.

[53] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1), 2000.

[54] E. Levitin and B. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 1966.

[55] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Applied Mathematics*, 16(6):964–979, 1979.

[56] S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Équations aux Dérivées Partielles*, pages 87–89, Paris, 1963. Éditions du centre National de la Recherche Scientifique.

[57] S. Łojasiewicz. Sur la géométrie semi- et sous- analytique. *Annales de l'institut Fourier*, 43(5):1575–1595, 1993.

[58] B. Martinet. Brève communication. régularisation d'inéquations variationnelles par approximations successives. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 4(3):154–158, 1970.

[59] 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.

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

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

[62] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[63] M. Nikolova. Markovian reconstruction using a gnc approach. *Image Processing, IEEE Transactions on*, 8(9):1204–1220, 1999.

[64] 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.

[65] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006.

[66] P. Ochs, T. Brox, and T. Pock. ipiasco: Inertial proximal algorithm for strongly convex optimization. *Technical Report*, 2014.

[67] P. Ochs, Y. Chen, T. Brox, and T. Pock. ipiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[68] P. Ochs, A. Dosovitskiy, T. Pock, and T. Brox. An iterated $\ell_1$ algorithm for non-smooth non-convex optimization in computer vision. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[69] 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, Nov. 1987. IEEE Computer Society Press.

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

[71] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *International Conference on Computer Vision (ICCV)*, 2009.

[72] 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.

[73] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[74] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1969.

[75] R. T. Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Applied Mathematics*, 14(5), 1976.

[76] R. T. Rockafellar. *Variational Analysis*, volume 317. Springer Berlin Heidelberg, Heidelberg, 1998.

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

[78] 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 (NIPS)*, pages 539–547, 2012.

[79] E.-G. Talbi and T. Muntean. Hill-climbing, simulated annealing and genetic algorithms: a comparative study and application to the mapping problem. In *System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on*, volume ii, pages 565–573 vol.2, 1993.

[80] H. L. Thi, V. Huynh, and T. P. Dinh. Convergence analysis of dc algorithm for dc programming with subanalytic data. Technical report, Ann. Oper. Res., INSA-Rouen, 2009.

[81] 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.

[82] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013.

[83] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010.

[84] A. J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996.

[85] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012.

[86] S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.