Escaping Poor Local Minima in Large Scale Robust Estimation
Generalized Majorization-Minimization and Filter Methods

Huu Le · Christopher Zach

Received: date / Accepted: date

Abstract Robust parameter estimation is a crucial task in several 3D computer vision pipelines such as Structure from Motion (SfM). State-of-the-art algorithms for robust estimation, however, still suffer from difficulties in converging to satisfactory solutions due to the presence of many poor local minima or flat regions in the optimization landscapes. In this paper, we introduce two novel approaches for robust parameter estimation. The first algorithm utilizes the Filter Method (FM), which is a framework for constrained optimization allowing great flexibility in algorithmic choices, to derive an adaptive kernel scaling strategy that enjoys a strong ability to escape poor minima and achieves fast convergence rates. Our second algorithm combines a generalized Majorization Minimization (GeMM) framework with the half-quadratic lifting formulation to obtain a simple yet efficient solver for robust estimation. We empirically show that both proposed approaches show encouraging capability on avoiding poor local minima and achieve competitive results compared to existing state-of-the-art robust fitting algorithms.

Keywords Robust Fitting · Structure From Motion · Majorization Minimization

1 Introduction

Robust model fitting is a fundamental task in many computer vision problems such as SLAM or Structure-from-Motion (SfM) to discard a potentially large fraction of outliers that could severely impact the final estimates. In low-dimensional problems such as fundamental or essential matrix estimation, the use of RANSAC [11] and its variants [3,4,20] are usually sufficient. However, these randomized approaches are not applicable to large-scale high-dimensional problems such as Bundle Adjustment (BA, e.g. [31]). The most popular approach to add robustness to large-scale optimization problems (and therefore including BA) are M-estimators [17], which fit particularly well for non-linear least-squares instances. Within the M-estimator framework, different robust loss functions (kernels) are available such as Huber [17], Cauchy, Tukey bi-weight or Smooth Truncated Least Squares (TLS) [36]. While convex kernels such as Huber work relatively well for data with low outlier rates, quasi-convex kernels such as Tukey or TLS need to be employed for highly contaminated data. The use of these quasi-convex kernels, however, leads to highly non-convex optimization problems containing many sub-optimal local minima or flat regions. As a result, solving these problems to is often challenging since most algorithms are very likely to be trapped at a poor local minimum.

A number of algorithms have been proposed in the literature to tackle the high non-convexity of robust estimation, and [38] evaluates some of the promising methods. While Iteratively Re-weighted Least Squares (IRLS) is rather popular and easy to implement, it often converges to poor sub-optimal solutions. In contrast, graduated optimization, which is often referred as graduated non-convexity (GNC) in the computer vision community [3,38,13], shows to be the most promising approach due to its appealing ability to avoid undesirable solutions. Therefore, GNC-based algorithms have attracted more attention in many robust fitting applications (e.g. [3,21,38,14]). However, the use of GNC requires a careful design of the graduation (or annealing) schedule, hence prior knowledge about the problem is desirable. A wrong schedule may cause either unnecessarily long run time in easy problem instances, where basic techniques that
provide fast convergence such as IRLS are sufficient, or undesirable results as local minima are not effectively avoided (as demonstrated in Figure 4).

**Contributions** In this paper, we propose two novel algorithms that possess strong ability to escape poor local minima. In particular,

- The first proposed method leverages GNC and the Filter Method (FM) [12], which is commonly used in the optimization literature to solve constrained optimization problems, to devise a novel adaptive kernel scaling scheme. In contrast to the conventional GNC with a fixed graduated schedule, our new algorithm allows the robust kernel to be adaptively scaled, hence it achieves a faster convergence rate compared to GNC.

- Our second method is inspired by the idea of Generalized Majorization Minimization (GeMM) [26], which we extend to a relaxed variant termed “ReGeMM”. To leverage ReGeMM for robust fitting, we make use of the special properties of robust fitting with the half-quadratic (HQ) lifting formulation [36], resulting in a new weight update scheme that helps the solver to converge to better solutions.

In light of our specific algorithmic details, both methods can be interpreted as instances of graduated optimization with an annealing schedule that is driven by the internal working of the underlying optimization method instead of externally defining an annealing schedule. Both algorithms can be easily integrated to existing non-linear least squares solvers such as Ceres [11] or SSBA [36]. We conduct many experiments on several large-scale bundle adjustment instances and show that our algorithms offer competitive performance compared to state-of-the-art approaches. This paper extends our two separate published works on large-scale robust fitting [20,41], where a significant amount of experimental results on larger datasets are added. Our source code is released as an updated version of the standalone C++ SSBA library that collects many existing state-of-the-art algorithms for large-scale bundle adjustment.

**Paper Outline** The rest of this paper is structured as follows,

- In Section 2 we briefly review the literature and discuss several commonly used approaches for robust fitting.

- The problem formulation of robust estimation, background theory are discussed in Section 3. In addition, Section 4 elaborates the limitations of graduated non-convexity—the current method-of-choice for smooth but very difficult optimization instances.

- Our Adaptive Kernel Scaling method with Filter Method is discussed in Section 5.

- The second algorithm which is based on a generalization of the majorization-minimization framework is discussed in Section 6.

- Finally, in Section 7 we provide experimental results to benchmark the performance of our proposed methods.

2 Related Work

Iteratively Re-weighted Least Squares (IRLS [16]) is arguably the most popular method being used to optimize high-dimensional robust cost functions. The main idea behind this approach is to associate each measurement (or corresponding least-squares term in the overall objective) with a weight based on the current residual value, followed by weighted least-squares minimization to obtain a refined solution. The weights are updated after each iteration and the process repeats until convergence. It has been demonstrated that with a proper initialization of weights, IRLS may provide competitive results [40]. However, for more complex problems, the returned solutions are usually not satisfactory as it is very easy for IRLS to be trapped in a poor local minimum.

To address the non-convexity of robust estimation, Zach [36] proposed to leverage the half-quadratic minimization principle [12] and derive algorithms to solve the problem in a “lifted” domain, where the non-convex robust kernel is re-parameterized by a new function in a higher dimensional space. The reformulated robust estimation problem incorporates both the original parameters and newly introduced unknowns representing the confident weights of the measurements. By employing such lifting approach, the flat region in the robust kernels can be avoided by indirectly representing the robustness into the new lifted objective, which is less sensitive to poor local minima. Using the lifting mechanism, different formulations and schemes have also been introduced. In contrast to the Multiplicative Half- Quadratic (M-HQ) lifting approach proposed in [35], Additive Half-Quadratic (A-HQ) has also been introduced [15,39]. A double lifting method that combines M-HQ and A-HQ is also discussed in [39]. However, the above lifting approaches have some limitations. In particular, [40] demonstrates that the success of half-quadratic minimization relies on suitable initialization of confidence weights, and that M-HQ fails on problems with multiple “competing” residuals.

Besides lifting, another popular approach to tackle problems containing many poor local minima is to “smooth” the objective using homotopy or graduation techniques [28,9,21] such as Graduated Non-convexity (GNC [3]). The underlying concept of graduated optimization is to successively approximate the original

\[1\text{ https://github.com/intellhave/ROSBA}\]
non-convex cost function by surrogate functions that are easier to minimize (i.e., leading to fewer local minima). In robust cost optimization, the surrogate functions may be chosen as a scaled version of the original robust kernel (see [38], which induces fewer local minima than the original cost. Graduated optimization and GNC have demonstrated their utility in several large-scale robust estimation problems by guiding the optimization process to relatively good local minima compared to other approaches such as IRLS or lifting variants [38].

3 Background

3.1 Problem Formulation

In this work, we are interested in the large-scale robust estimation task under the M-estimators framework. Given a set of $N$ measurements, and let us denote the residual vector induced by the $i$-th observation by $r_i(\theta) \in \mathbb{R}^p$, where the vector $\theta \in \mathbb{R}^d$ contains the desired parameters. In robust cost optimization, we wish to obtain the optimal parameters $\theta^*$ that solve the following program

$$\theta^* = \arg\min_{\theta} \Psi(\theta) \quad \Psi(\theta) := \sum_{i=1}^{N} \psi(||r_i(\theta)||),$$

where $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is a symmetric robust kernel that satisfies the following properties [14,38]: $\psi(0) = 0$, $\psi''(0) = 1$, and the mapping $\phi : \mathbb{R}^+_1 \rightarrow \mathbb{R}^+_0$ where $\phi(x) = \psi(\sqrt{2x})$ is concave and monotonically increasing.

The problem [1] serves as a generic framework for several robust fitting tasks, in which the definitions of the parameters $\theta$ and the residual vectors $\{r_i(\theta)\}$ depend on the specific application. For example, in robust metric bundle adjustment, the vector $\theta$ contains all the camera matrices $\{R_j,t_j\}_{j=1}^{N_c}$ and 3D points $\{X_k\}_{k=1}^{N_p}$ that we wish to estimate ($N_c$ and $N_p$ are the number of cameras and the number of points, respectively), and each residual vector $r_{ij} \in \mathbb{R}^2$ is defined as

$$r_{ij}(\theta) = u_{ij} - \pi(R_jX_j+t_j),$$

where $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is defined as $\pi(X) = (X_1/X_3,X_2/X_3)$, and $u_{ij}$ is the 2D keypoint corresponding to the $j$-th 3D point extracted in image $i$.

The robust kernel $\psi$ can be chosen from a wide range of functions (see [38]). This choice usually affects the robustness and the non-convexity of the resulting optimization problem. For example, if $\psi(x)$ is chosen such that $\psi(x) = \frac{1}{2}x^2$, one obtains the non-robust least squares estimate, which is straightforward and efficient to optimize but at the same time highly sensitive to outliers. In this work, if not otherwise stated, we chose $\psi$ to be the smooth truncated kernel,

$$\psi(r) = \begin{cases} \frac{1}{2}r^2 \left(1 - \frac{r^2}{\tau^2}\right) & \text{if } r^2 \leq \tau^2, \\ \tau^2/4 & \text{otherwise}. \end{cases}$$

This choice is a smooth and numerically convenient approximation to a highly robust but non-smooth truncated quadratic kernel.

3.2 Filter Method for Constrained Optimization

The filter method [12] was initially developed as an alternative to penalty approaches for constrained optimization [25]. In order to outline the filter method, let us consider a general constrained optimization problem,

$$\min_{x \in \mathbb{R}^d} f(x), \text{ s.t. } g_i(x) = 0, i = 1 \ldots c,$$

where $f,g_i : \mathbb{R}^d \rightarrow \mathbb{R}$ are continuously differentiable functions, while $c$ is the number of constraints. We also introduce a function $h(x)$ quantifying constraint violations. A typical choice for $h$ (which we use in the following) is given by $h(x) = \sum||g_i(x)||$. Clearly, $h(x^*) = 0$ iff $x^*$ is a feasible solution of [4]. In classical penalty approaches, the constraint violation is incorporated into the objective with a penalty parameter $\mu$ in order to obtain a new but unconstrained objective (i.e., $f(x) + \mu h(x)$). The resulting objective can then be optimized using a suitable local method. Usually, $\mu$ increased monotonically according to a specified schedule to ensure that the solution converges to a feasible region of [4]. One drawback of such approach is that the initial value of $\mu$ and how it is increased must be carefully tuned. Another practical issue with penalty methods is, that feasibility of the solution is only guaranteed when $\mu \rightarrow \infty$ (unless one utilizes an exact but usually non-smooth penalizer $h$ [25]).

In contrast to penalty methods, Fletcher et al. [12] proposes an entirely different mechanism to solve [4] by introducing the concept of a filter (see Figure 1), which offers more freedom in the step computation. At a current value of $x$, let us denote by $F(x)$ the pair combining the objective value and the associated constraint violation, $F(x) = (f(x),h(x)) \in \mathbb{R}^2$. For brevity, we sometime use $f$ and $h$ to denote $f(x)$ and $h(x)$, respectively. Given two pairs $F_i = (f_i,h_i)$ and $F_j = (f_j,h_j)$, the concept of domination is defined as follows: $F_i$ is said to dominate $F_j$ if $f_i < f_j$ and $h_i < h_j$. A filter is then defined as a set $\mathcal{F} = \{F_j\}_{j=1}^{\infty} \subseteq \mathbb{R}^2$ containing mutually non-dominating entries. The filter $\mathcal{F}$ defines a dominated (and therefore forbidden) region $\mathcal{F}$ in the 2D plane. A pair $F_i$ is said to be accepted by the filter $\mathcal{F}$ if it is not dominated by any pair in the set $\mathcal{F}$. This method has been successfully applied to various large-scale optimization problems [25].
**Algorithm 1** Optimization with Filter Method

**Require:** Initial solution \(x^0\), filter margin \(\alpha, \max\_\text{iter}\)

1: Initialization: \(t \leftarrow 0, \mathcal{F} \leftarrow \emptyset\)

2: **while** true and \(t < \max\_\text{iter} \) **do**

3:  **if** \(x^t\) is stationary **then**

4:    **break**;

5:  **end if**

6:  \(f \leftarrow f_t - \alpha h_t; h_t \leftarrow h - \alpha h_t\)

7:  \(\mathcal{F} \leftarrow \mathcal{F} \cup \{(f, h)\}\)

8:  \(F_{t+1} \leftarrow \{x|f(x) \geq f, h(x) \geq h\}\)

9:  \(\mathcal{F} \leftarrow \mathcal{F} \cap F_{t+1}\)

10: **Compute** \(x^{t+1} \notin \mathcal{F}\) (Sec. 5.1.1 and 5.1.2)

11: **if** \(f(x^{t+1}) < f(x^t)\) **then**

12:    \(\mathcal{F} \leftarrow \mathcal{F} \setminus \{(f, h)\}; \mathcal{F} \leftarrow \mathcal{F} \cap F_{t+1}\)

13: **end if**

14: **end while**

15: **return** \(x^t\)

---

**Figure 1:** Example of a filter. The \(x\) axis depicts the main objective \(f\), while the \(y\) axis represents the constraint violation \(h\). The gray area indicates the forbidden region defined by three mutually non-dominated pairs (shown in red). Optimization with filter method involves finding, from a current \(x^t\), a new value \(x^{t+1}\) that is not dominated by the filter. A step that reduces both \(f\) and \(h\) is preferable (as illustrated by the blue arrow).

\[ J(\theta, \mathbf{u}) \] is a differentiable objective, which is bounded from below. Further, \(\theta \in \mathcal{U} \subseteq \mathbb{R}^d\) denotes the complete set of latent variables. We assume w.l.o.g. that \(\tilde{J}(\theta, \mathbf{u}) \geq 0\) for all \(\theta\) and \(\mathbf{u}\). In robust fitting, we are interested in functions \(J(\theta)\) with the form:

\[
\min_{\theta} J(\theta) := \min_{\theta \in \mathcal{U}} \min_{\mathbf{u} \in \mathcal{W}} J(\theta, \mathbf{u}),
\]

where \(J: \mathbb{R}^d \times \mathcal{W} \rightarrow \mathbb{R}_{\geq 0}\) is a differentiable objective function, that is bounded from below. Further, \(\theta \in \mathcal{U} \subseteq \mathbb{R}^d\) denotes the complete set of latent variables. We assume w.l.o.g. that \(\tilde{J}(\theta, \mathbf{u}) \geq 0\) for all \(\theta\) and \(\mathbf{u}\). In robust fitting, we are interested in functions \(J(\theta)\) with the form:

\[
\min_{\theta} J(\theta) := \frac{1}{N} \sum_{i=1}^N J_i(\theta, \mathbf{u}_i),
\]

where \(\theta\) are the main parameters of interest and \(\{\mathbf{u}_i\}\) are e.g. the explicit confidence weights used to model robust kernels. Observe that, by construction, the function \(\tilde{J}(\theta, \mathbf{u})\) provides an upper bound to \(J(\theta)\). Also, we assume that for given \(\theta\), the task of determining the minimizer \(\arg\min_{\mathbf{u} \in \mathcal{W}} J(\theta, \mathbf{u})\) can be done efficiently. In summary, in our setting, the following properties hold:

1. \(J(\theta, \mathbf{u}) \geq J(\theta)\) for all \(\theta \in \mathbb{R}^d\) and \(\mathbf{u} \in \mathcal{W}\).
2. \(J(\theta, \mathbf{u})\) is convex in \(\mathbf{u}\) and satisfies strong duality.
3. \(J(\theta) := \min_{\mathbf{u} \in \mathcal{W}} J(\theta, \mathbf{u})\).

---

3.3 Majorization Minimization and Relaxed Variants

Majorization Minimization (MM) \cite{huu2012majorization} is an often employed optimization paradigm for optimizing a wide range of optimization problems. The main idea behind MM is to solve the original problem by iteratively optimizing a typically convex surrogate function, for which solutions can be easily obtained. This approach generalizes other methods such as expectation-maximization (EM) \cite{dempster1977maximum} and the convex-concave procedure \cite{wright1999numerical}. In the following, we briefly review MM and its generalized variant, which serves as the background for our proposed method discussed in Section 6.

We consider the task of determining a stationary point \(\theta^*\) of an objective \(J\), where \(J\) itself involves optimization over additional latent variables,

\[
\min_{\theta} J(\theta) := \min_{\theta \in \mathcal{U}} \min_{\mathbf{u} \in \mathcal{W}} J(\theta, \mathbf{u}),
\]

since convergence to true (local) minimizers is difficult to guarantee in the general non-convex setting.
In conventional MM, at each iteration, the latent variables are optimized such that the following “touching” condition is satisfied,
\[
J(\theta(t-1),u(t)) = J(\theta(t-1), \bar{u}(t-1)).
\] (7)

By alternatively optimizing \(u\) and \(\theta\) such that the touching condition (7) is satisfied, the convergence to a local solution is guaranteed.

In contrast to standard MM, generalized MM [26], replaces the touching condition to a “sufficient decrease” criterion,
\[
J(\theta(t-1), \bar{u}(t)) \leq \eta J(\theta(t-1)) + (1 - \eta) J(\theta(t-1), \bar{u}(t-1))
\]
\[
= J(\theta(t-1), \bar{u}(t-1)) - \eta (J(\theta(t-1), \bar{u}(t-1)) - J(\theta(t-1))),
\] (8)

where \( \eta \in (0,1) \) is a user-specified parameter. By construction the gap \( d_t := J(\theta(t), u(t-1)) - J(\theta(t-1)) \) is non-negative. The above condition means that \( \bar{u}(t) \) has to be chosen such that the new objective value \( J(\theta(t), \bar{u}(t)) \) is guaranteed to sufficiently improve (but not necessarily more) over the current upper bound \( J(\theta(t-1), u(t-1)) \).

\[
J(\theta(t), \bar{u}(t)) \leq J(\theta(t-1), \bar{u}(t)) \leq J(\theta(t-1), u(t-1)) - \eta d_t.
\]

It is shown that the sequence \( \lim_{t \to \infty} d_t \to 0 \), i.e. asymptotically the main objective \( J \) of interest is optimized. Since generalized MM decreases the upper bound less aggressively than standard MM, it has an improved empirical ability to reach better local minima in highly non-convex problems [26].

The condition in Eq. 8 can be further relaxed to [41]
\[
J(\theta(t-1), \bar{u}(t)) \leq \eta J(\theta(t-1)) + (1 - \eta) J(\theta(t-2), \bar{u}(t-1)).
\] (9)

The advantage of the condition in Eq. 9 over Eq. 8 is that the latent variables \( \bar{u}(t-1) \) can be immediately discarded once the new solution \( \theta(t-1) \) is determined, and that an additional evaluation of the full cost \( J(\theta(t-1), \bar{u}(t-1)) \) is avoided. Otherwise similar guarantees hold as with generalized MM [41]. We call the algorithm based on Eq. 9 “relaxed generalized MM” (or ReGeMM for short) and refer to Fig. 2a for an illustration of the difference between generalized MM and ReGeMM.

4 Graduated Optimization and Its Limitations

In this section, we briefly review graduated optimization (or graduated non-convexity [13]), which is a popular technique commonly employed to avoid poor local minima in highly non-convex problems. The limitations discussed in this section also serve as the motivation for our novel methods proposed in this paper. Indirectly, graduated methods are also leveraged in coarse-to-fine schemes used e.g. in variational methods for optical flow [22]. The main
The idea behind this technique is to optimize the original highly non-convex cost function $\Psi$ by minimizing a sequence of problems $(\Psi^k, \ldots, \Psi^0)$, where $\Psi^0 = \Psi$ and $\Psi^{k+1}$ is “easier” to optimize than $\Psi^k$. Starting from the original robust kernel $\Psi$ (as defined in (1)), the set of “easier” problems are obtained by a scaled version of $\Psi$. In particular, from the original minimization problem with the objective function $\Psi(\theta)$, each problem $\Psi^k$ is constructed with a new kernel $\Psi^k$:

$$\Psi^k(r) = s_k^2 \Psi\left(\frac{r}{s_k}ight),$$

(10)

where the scale parameters are chosen such that $s_{k+1} > s_k$ and $s_0 = 1$. Figure 3 shows an example of a one dimensional robust mean estimation, where we plot the objective values of the problem with the original kernel and its scaled version (with $s = 3$). As can be seen, the scaled kernel results in this case in a problem with a single global minimum.

![Figure 4: A wrong schedule of GNC may lead to either poor results (GNC-2, which is not better than IRLS) or unnecessary iterations (GNC-5). Here GNC-2 and GNC-5 mean GNC with the number of levels $k$ set to 2 and 5, respectively. Our proposed method provides competitive objective value and converges faster than GNC.](image)

To the best of our knowledge, methods that rely on graduated optimization achieve state-of-the-art results for large-scale robust estimation tasks (most importantly, bundle adjustment problems) due to their ability to escape poor local minima. However, in practice it is necessary to define a schedule with a fixed number of levels $k$. This requires some knowledge about the problem so that a proper value for $k$ can be assigned. A large value of $k$ may cause unnecessary iterations, which translates to high running time. On the other hand, setting a low $k$ may not provide sufficient scaling levels for the optimizer to avoid poor solutions (as shown in Figure 4). Moreover, in some easy applications, although GNC converges to a lower objective than its competitor (e.g., IRLS), the difference between the converged objectives may be insignificant. In such scenarios, an IRLS solver can provide acceptable results within a few iterations, while it may take longer for a GNC solver to go through all $k > 1$ levels. However, using IRLS poses a risk of converging to bad local minima. Therefore, there is a trade-off between the selecting a solver and associated hyper-parameters (such as the annealing schedule in GNC) and the resulting efficiency.

5 Adaptive Kernel Scaling

In this section, we describe our novel solver for robust parameter estimation that aims to leverage the advantages of GNC while at the same time avoids the weaknesses listed in Section 4. However, unlike previous graduated schemes employing a fixed schedule of kernel scaling, we consider the scale of each residual as a variable, and allow the scales to be jointly optimized with the set of parameters $\theta$. This leads us to a new formulation for robust estimation, which is a constrained optimization problem and can be written as

$$\min_{\theta, \{\sigma_i\}} \sum_{i=1}^{N} \Psi\left(\frac{\|r_i(\theta)\|}{\sigma_i}\right) \quad \text{s.t.} \quad \sigma_i = 1 \quad \forall i = 1, \ldots, N.$$  

(11)

In contrast to e.g. graduated optimization, which maintains usually a single smoothness parameter, we introduce a scaling factor $\sigma_i$ for each residual. Consequently, each scale $\sigma_i$ evolves differently during the optimization process. Clearly, (11) does not appear helpful, as enforcing the constraints $\sigma_i = 1$ strictly (i.e. maintaining a feasible solution throughout) makes (11) equivalent to the original task (1). Strategies such as graduated optimization do not maintain strictly feasible iterates, but use a schedule for $\sigma_i$ to eventually satisfy the constraints. Turning the original problem (1) into a constrained optimization problem (11) has two potential benefits: first, a larger set of optimization methods is applicable, and second, intermediate solutions may be infeasible but at the same time correspond to smoother problem instances.

Observe that in order to obtain a solution for (11), besides the initialization $\theta_0$ for the parameters, one can also initialize the scales $\sigma_i$ to values that are greater than 1 and expect that the solver will drive $\sigma_i$ to the feasible region $\sigma_i = 1$ of (11). Therefore, by considering the problem (11) and setting $\sigma_i$ to initial values greater than 1, we are effectively conducting kernel scaling, which provides the potential of escaping poor local minima. In contrast to graduated optimization, the internal workings of the optimization method determine how feasibility of $\sigma_i$ is eventually achieved. In particular, $\sigma_i$ may be updated in non-monotonically and therefore being increased during the iterations of the optimization method. In this work we propose to utilize a filter method to address the constrained problem (11), since it is a highly flexible and non-monotone framework for constrained optimization problems.

See Appendix A for the discussion on an alternative filter formulation.
5.1 Optimization with Filter Method

By introducing the scale variables \( \{ \sigma_i \} \), we obtained a constrained optimization problem as written in (11). One requirement for the optimization method of choice is that the limit values of \( \sigma_i \) must be 1 when the algorithm converges. Moreover, any proposed method for solving (11) should be competitive with existing second-order solvers for problem instances (1) (such as Ceres [11] and SSBA [36]). This requirement rules out e.g. first order methods for constrained programs.

Our approach to solve (11) follows closely the steps described in Algorithm 1. However, the main contribution of our work is a novel strategy to compute \( x^{t+1} \) that is accepted by the filter. In addition, our method is able to leverage existing non-linear least-squares solvers.

We restrict \( \sigma_i \) to be greater or equal to 1, as \( \sigma_i \in (0, 1) \) will lead to a harder problem than (1). Therefore, it is convenient to re-parameterize \( \sigma_i \) as \( \sigma_i = 1 + s_i^2 \) and we can rewrite the problem (11) as follows:

\[
\min_{\theta, \{ s_i \}} \sum_{i=1}^{N} \psi \left( \frac{r_i(\theta)}{1 + s_i^2} \right) \quad \text{s.t.} \quad s_i = 0 \quad \forall i. \tag{12}
\]

In the context of (4), let \( x = [\theta^T, s^T]^T \) where \( s = [s_1 \ldots s_0]^T \) is a vector that collects the values of \( s_i \). Finally, the functions \( f(x) \) and \( h(x) \) correspond to

\[
f(x) = \sum_{i=1}^{N} \psi \left( \frac{r_i(\theta)}{1 + s_i^2} \right) \quad h(x) = \sum_{i} s_i^2. \tag{13}
\]

5.1.1 Cooperative Step

An appealing feature of Algorithm 1 is that it offers a flexible choice of algorithms to perform variable update, as long as \( x^{t+1} \) is accepted by the filter (i.e., \( x^{t+1} \notin \mathbb{F} \) as described in Line. 10 of Algorithm 1). Like filter methods for non-linear constrained minimization there are two possible steps to obtain a new acceptable iterate: the cooperative step described in this section is the main workhorse of the algorithm. It replaces the sequential quadratic program (SQP) used as the main step in filter methods for general non-linear programs [12][27]. The cooperative step is complemented with a restoration step as a fall-back option, that is described in the following section.

The cooperative step is motivated by the fact that reducing both the main objective and the constraint violation (i.e., \( f(x) \) and \( h(x) \)) by a sufficient amount (as induced by the margin parameter \( \alpha \)) leads to a new solution that is guaranteed to be acceptable by the filter. We use a second-order approximation of \( f \) and \( h \) around the current values \( x' \),

\[
f(x' + \Delta x) = f(x') + g_f^T \Delta x + \frac{1}{2} \Delta x^T H_f \Delta x,
\]

\[
h(x' + \Delta x) = h(x') + g_h^T \Delta x + \frac{1}{2} \Delta x^T H_h \Delta x,
\]

(14) where \( g_f \) and \( g_h \) are the gradients, while \( H_f \) and \( H_h \) are true or approximated Hessian of \( f \) and \( h \), respectively. Hence, a cooperative update direction \( \Delta x \) possibly decreasing both \( f \) and \( h \) is given by (13).

\[
\arg \min_{\Delta x} \{ f, h \} \quad \text{where} \quad \Delta f = g_f^T \Delta x + \Delta x^T H_f \Delta x
\]

\[
\Delta h = g_h^T \Delta x + \Delta x^T H_h \Delta x.
\]

(16) This is a convex quadratic program, which can be efficiently solved using any iterative solver. However, as previously discussed, our ultimate goal is to integrate our algorithm into existing solvers: following [40], instead of solving (15) the update \( \Delta x' \) is obtained via a relaxed problem,

\[
\Delta x' = \arg \min_{\Delta x} \mu_f f + \mu_h h, \tag{17}
\]

where \( \mu_f > 0 \) and \( \mu_h > 0 \) with \( \mu_f + \mu_h = 1 \) are suitably chosen coefficients. Adding a Levenberg-Marquardt-type damping [23] with parameter \( \lambda \) yields

\[
\Delta x' = (\mu_f H_f + \mu_h H_h + \lambda I)^{-1}(\mu_f g_f + \mu_h g_h).
\]

(18) If the new iterate \( x^{t+1} = x' + \Delta x' \) is acceptable by \( \mathbb{F} \), then \( \lambda \) is decreased, otherwise increased.

With an appropriate choice of \( \mu_f \), \( \mu_h \) and a sufficiently large \( \lambda \), it can be shown that \( \Delta x' \) leads to a reduction of both \( f \) and \( g \) as long as \( g_f \) and \( g_f \) are not pointing in opposite directions [40]. If \( x' + \Delta x' \) leads to a sufficient decrease of both \( f \) and \( h \), then this new solution is by construction acceptable by the current filter. Otherwise, the new iterate may be still acceptable, but increases either \( f \) or \( h \) (and is therefore a non-monotone step). If the new solution is not acceptable by the filter, then a non-monotone restoration step is applied (that also leads to an increase of either \( f \) or \( h \)). The filter condition ensures that \( h \) eventually converges to 0. We set \( \mu_f \) to 0.9 and \( \mu_h \) to 0.1 for all datasets tested in our experiments.

5.1.2 Restoration Step

Although (18) gives us a way to compute preferable update step, it does not guarantee to provide always steps that are accepted by the filter. In such cases, we revert to a restoration step described below.

In the filter methods literature a restoration step essentially reduces the constraint violation and is applied if the SQP step did not yield an acceptable new iterate. Note
that in our setting, just reducing the constraint violation is trivial, and a perfectly feasible solution can be obtained by setting \( s_i = 0 \) for all \( i \). A good restoration step aims to yield a good starting point for the next main step (which is SQP in traditional filter methods and a cooperative step in our approach). Consequently, the goal of our restoration step is to determine a suitable new solution for the subsequent cooperative step. One simple criterion for such a new point is given by the angle between the gradients of \( f \) and \( h \), which is to be minimized in order to facilitate cooperative minimization. Our deliberate design choice is to adjust only the parameters \( s_i \) in the restoration step, i.e.

\[
\Delta x = \gamma \left( \begin{array}{c}
0 \\
\Delta s
\end{array} \right),
\]

where \( \gamma \) is a step-size determined by a grid search,

\[
\gamma = \arg \min_{\gamma} \angle (g_f(x + \Delta x), g_h(x + \Delta x)).
\]

Note that adjusting \( s \) affects both \( g_f \) and \( g_h \). The search direction \( \Delta s \) is chosen as \( \Delta s = -s \). Due to the particular choice of \( h \) this search direction coincides with the direction to the global minimum \( s = 0 \) of \( h \), with the negated gradient \(-\nabla_s h(s)\), and with a Newton step optimizing \( h \). We limit the search for \( \gamma \) to the range \([0, 1/2]\). The detailed computations of the update steps are summarized in Algorithm 2.

### Algorithm 2 Step computations in Adaptive Kernel Scaling (ASKER)

**Require:** Current value \( x' \), \( \mu_f, \mu_h \), current damping value \( \lambda' \).

Compute \( g_f, g_h, H_f, H_h \) from \( x' \).

Compute \( \Delta x \) using (13).

\[ x^{i+1} \leftarrow x' + \Delta x \]

if \( x^{i+1} \notin \mathbb{R}^n \) then

\(*\)/Perform Restoration Step*/

\[ \Delta x \leftarrow \gamma \left( \begin{array}{c}
0 \\
-\Delta s
\end{array} \right), \gamma \text{ is computed based on (15),} \]

\[ x^{i+1} \leftarrow x' + \Delta x \]

\[ \lambda^{i+1} \leftarrow 10\lambda' \]

else

\[ \lambda^{i+1} \leftarrow \lambda'/10 \]

end if

return \( x' \)

---

### 6 Relaxed Generalized MM for Robust Fitting

We leverage the ReGeMM framework presented in Section 3.3 in order to derive a new algorithm for robust fitting. First, following the half-quadratic lifting formulation, the robust parameter estimation can be rewritten as follows

\[
\min_{\theta, \pi} J(\theta, \pi), \text{ with } J(\theta, \pi) := \sum_{i=1}^{N} \left( \frac{\pi_i}{2} \| r_i(\theta) \|^2 + \kappa(\pi_i) \right),
\]

where \( \pi_i \) here acts as the confident weight for the \( i \)-th residual, and the function \( \kappa(\cdot) : [0, 1] \rightarrow \mathbb{R}_{\geq 0} \) is a convex and monotonically decreasing function that serves as a “bias” function. The exact shape of \( \kappa \) depends on the robust kernel \( \psi \). For instance, using the smooth truncated least squares introduced in Eq. 37, the function \( \kappa \) can be derived as,

\[
\kappa(u) = \frac{u^2}{4}(u - 1)^2,
\]

where \( u \geq 0 \) is a confidence weight. In can be shown that \( \min_{u \geq 0} u^2/2 + \kappa(u) = \psi(|r|) \), and we refer to e.g. [14, 36] for more details on the connection between \( \kappa \) and \( \psi \).

Observe that the lifting formulation as written in (21) is a special instance of [3], where the latent variables are the confident weights \( \pi_i \).

At the \( t \)-th iteration, given the current value of \( \theta^{(t-1)} \), the conventional IRLS algorithm updates each confident weight \( \pi_i^{(t)} \) by solving

\[
\pi_i^{(t)} = \arg \min_{\pi} \left\| r_i(\theta^{(t-1)}) \right\|^2 + \kappa(\pi_i).
\]

The value of \( u_i^* \) that solves (23) can be computed in closed form using the weight function \( \omega \)

\[
\pi_i^{(t)} = \omega(\| r_i \|), \text{ with } \omega(x) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} := \psi'(x)/x.
\]

Following the ReGeMM framework, the weights in our new algorithm are only “partially” updated such that the ReGeMM criterion (Eq. 9) holds,

\[
J(\theta^{(t-1)}, \pi^{(t)}) \leq \eta J(\theta^{(t-1)}) + (1 - \eta) J(\theta^{(t-2)}, \pi^{(t-1)}),
\]

The rationale is that the confidence weights are not fully committed to the current value of the residual, but overall convergence is still guaranteed. Since \( \kappa \) is a monotonically decreasing function, the new weights can be updated using a bisection approach. In particular, we set \( \pi_i^{(t)} = \psi(\frac{1}{\sigma}) \| r_i^{(t-1)} \| \) and perform bisection on \( \sigma \) until the criterion (25) is satisfied. Thus, we search for the largest \( \sigma \) satisfying Eq. 25 and the ReGeMM criterion induces an optimization-driven schedule for the scaling parameter \( \sigma \).

After updating the confident weights, the new fitting parameters are obtained by solving the weighted non-linear least-squares problem,

\[
\theta^{(t)} = \arg \min_{\theta} \sum_{i=1}^{N} \pi_i^{(t)} \| r_i^{(t-1)}(\theta) \|^2.
\]

In our implementation we use the Levenberg-Marquardt method, and \( \theta^{(t)} \) is the value obtained after one successful Levenberg-Marquardt iteration, thereby guaranteeing a sufficient decrease of \( J(\cdot, \pi^{(t)}) \). The algorithm (ReGeMM) is summarized in Algorithm 3.
7 Experimental Results

In this section, we provide several experimental results to demonstrate the performance of our algorithms and compare them against current state-of-the-art approaches. While our algorithms can be straightforwardly applied to most robust model fitting applications in computer vision, we use robust bundle adjustment (BA) (with the residual function described in Eq. (2)) as the main problem of interest for our experiments, as it is consider one of the most challenging tasks due to its high dimensionality. We benchmark the algorithms using the following publicly available datasets:

- Bundle Adjustment in the Large (BAL) \cite{2}. This is a well-known dataset provided by \cite{2} for bundle adjustment experiments.\(^3\) This dataset contains the 3D structures reconstructed from a set of images as described in \cite{2}. The whole reconstruction is divided into five sub-datasets: Ladybug, Trafalgar Square, Dubrovnik, Venice, and Final. We extract sequences that are considered challenging for robust estimation (See the corresponding figures for the name of the selected sequences). We conduct metric bundle adjustment that optimizes the camera poses and the 3D points, with the residual function as described in Eq. (2). For brevity, we denote each instance by \(D - c\), where \(D\) is the first letter of the dataset name (e.g., F for Final) and \(c\) is the number of images (cameras).

- Bundle Fusion \cite{6} Dataset: In order to test our algorithms on the robust dense BA application, which is useful in e.g., real-time SLAM applications that often conduct local BA, we extracted 50 frames from the apt0 and office0 sequences (starting from frame 200), and tracked the associated 3D points via the provided depth maps and camera poses.

- 1dsfm: We also test the performance of our algorithms on several instances of the 1dsfm datasets \cite{32}: New York Library, Montreal Notre Dame, and Tower of London. We use the TheiaSfm library \cite{29} to obtain the initial reconstructions, and use our algorithms to perform robust BA.

We compare our algorithms (i.e., ASKER and ReGeMM) against commonly used methods for large-scale robust fitting, including IRLS \cite{15}, Multiplicative Half-Quadratic Lifting (M-HQ) \cite{36}, Graduated Non-Convexity (GOM+) as implemented in \cite{38}, and LM-MOO \cite{40}. Note that for GNC, we use the early stopping criterion introduced in \cite{38}, which allows GNC to achieves faster convergence rates compared to the fixed schedule. To make the notations consistent with previous work \cite{38}, we use GOM+ to denote GNC throughout all experiments.

We implement our algorithm in C++ using the framework provided by SSBA\(^4\), which is originally based on direct sparse linear solver\(^5\). In order to evaluate large-scale BA instances, we replaced the direct solver with a conjugate gradient implementation using a block-diagonal preconditioner. The stopping criterion for PCG iterations is the same as the one given in \cite{2} (i.e. forcing sequence parameter equal to \(1/10\) and a maximum of 1000 iterations). All experiments are executed on an Ubuntu workstation with an AMD Ryzen 2950X CPU and 64GB RAM. Other methods are also implemented based on the SSBA framework. For better visualization of the figures, we only compare our algorithm against the methods listed above, which are the best representatives for baseline and state-of-the-art approaches. As reported in \cite{38}, methods such as square-rooting the kernels \cite{10} or Triggs correction \cite{31} do not offer much improvement compared to IRLS, hence we omit these in our experiments. All methods are initialized from the same starting point. For ASKER, we set \(\mu_i\) to 0.9 and \(\mu_0\) to 0.1 for all experiments. The filter margin \(\alpha\) is set to 0.01, and initial scales \(s_i\) are set to 5. For ReGeMM, the value of \(\eta\) used in the the relaxed condition (25) is set to 0.5 for all experiments.

In Figure 5, we show the final inlier rates obtained by the methods after 50 iterations (using the inlier threshold of 1 pixel). Observe that ASKER, ReGeMM and GOM+ are the winning methods for most of the datasets. However, ASKER and ReGeMM perform better than GOM+ in several instances (e.g., V-427, L-138).

To summarize the performance of the algorithms throughout all datasets, we use the performance profiles \cite{8}. A performance profile indicates for each method the fraction \(\rho\) of problem instances, for which the method is within a factor \(\tau\) compared to the best method (with respect to a chosen performance measure). In Figure 6 (left) shows the performance profile w.r.t. to the best objective value reached after 50 iterations, while

---

\(^3\) The datasets can be downloaded at https://grail.cs.washington.edu/projects/bal/

\(^4\) https://github.com/chszach/SSBA

\(^5\) http://www.suitesparse.com
In Fig. 5, we depict the inlier rates obtained by the methods after 50 iterations. (a) Objectives (b) Inlier Rates

Figure 6 depicts the performance profiles for the best inlier rates obtained by the methods. As can be seen, both ASKER and ReGeMM can quickly reach satisfactory solutions, which justify their applicability in applications that require real-time performance.

To provide an in-depth analysis on the performance of the algorithms, we plot in Figure 7 the best objectives (normalized) for all methods after 100 iterations for 12 selected instances in the BAL datasets (We intentionally selected instances having more than 100 cameras for the experiments). As can be seen, ASKER and ReGeMM yield competitive results compared to GOM, LM-MOO and M-HQ. ReGeMM, in a large number of instances, shows to be the most competitive approach as it converges relatively fast to solutions that are very close to the optimal. Figure 8 shows similar plots for 3 selected instances in the 1dsfm dataset, and Figure 9 shows the results for two dense BA instances in the Bundle Fusion dataset. Observe that ReGeMM consistently yield very fast convergence rates.

8 Conclusion and Future Work

We introduced two new algorithms for large-scale robust fitting. The first method utilizes the Filter Method to derive an adaptive kernel scaling approach, while the second algorithm leverages a relaxed variant of the majorization-minimization principle, resulting in new algorithms that possess strong ability to escape poor local minima. Our algorithms can be easily integrated into existing sparse non-linear least squares solvers. Experiments on several large-scale datasets show that both algorithms achieve competitive performances with much faster convergence rates compared to existing robust solvers.

In our future work, several novel exploration strategies for the filter can be considered. Besides, one can also consider combining the ReGeMM condition with several incremental or stochastic optimization schemes to achieve even faster convergence rates for large-scale problems.

A Additive Lifting and the Filter Method

A different way to make robust estimation amenable to the filter method is by replicating residuals and enforce the overall consistency, e.g.

\[
\min_{\theta} \sum_{i} \psi(||p_i||) \quad \text{s.t.} \quad p_i = r_i(\theta).
\]

Such reformulation bears strong resemblance with the additive lifting formulation for robust costs [15] (which can be seen as introducing a quadratic penalizer for the constraint \( p_i = r_i(\theta) \)). The resulting program can also be solved using the Filter Method introduced in Algorithm 1. In particular, we define the objective \( f(x) \) and the constraint violation \( h(x) \) to be

\[
f(x) = \sum_{i} \psi(||p_i||) \quad h(x) = \sum ||p_i - r_i(\theta)||^2.
\]
Fig. 7: Performance of the tested algorithms: evolution of the objective versus wall clock time. We compare the proposed methods (ASKER and ReGeMM) against standard IRLS, M-HQ [36], GOM+ [38], and LM-MOO [40].

for two example instances in the BAL dataset. Observe that ASKER offers much better performance compared to the results using (27).

This empirical results conform with the observations discussed in previous work [39], where solving robust fitting using the additive lifting formulation offer inferior results compared to the half-quadratic (M-HQ) counterpart.
Fig. 8: Performance of the algorithms for the selected instances in the 1dsfm dataset.

Fig. 9: Performance of the algorithms for two selected instances in the Bundle Fusion dataset.

Fig. 10: Comparison between ASKER and the additive lifting based formulation in Eq. (27).

References

1. Agarwal, S., Mierle, K., Others: Ceres solver. http://ceres-solver.org
2. Agarwal, S., Shvetye, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: European conference on computer vision, pp. 29–42. Springer (2010)
3. Blake, A., Zisserman, A.: Visual reconstruction (1987)
4. Chum, O., Matas, J.: Matching with prosac-progressive sample consensus. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 220–226. IEEE (2005)
5. Chum, O., Matas, J., Kitten, J.: Locally optimized ransac. In: DAGM. Springer (2003)
6. Dai, A., Nieder, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) 36(4), 1 (2017)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39(1), 1–22 (1977)
8. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical programming 91(2), 201–213 (2002)
9. Dunlavy, D.M., O’Leary, D.P.: Homotopy optimization methods for global optimization. Tech. rep., Sandia National Laboratories (2005)
10. Engels, C., Stewénius, H., Nister, D.: Bundle adjustment rules. Photogrammetric computer vision 2 (2006) (2006)
11. Fischer, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
12. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Mathematical programming 91(2), 239–269 (2002)
13. Fliege, J., Drummond, L.G., Svaiter, B.F.: Newton’s method for multiobjective optimization. SIAM Journal on Optimization 20(2), 602–626 (2009)
14. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Transactions on pattern analysis and machine intelligence 14(3), 367–383 (1992)
15. Geman, D., Yang, C.: Nonlinear image recovery with half-quadratic regularization. IEEE transactions on Image Processing 4(7), 932–946 (1995)
16. Green, P.J.: Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. Journal of the Royal Statistical Society: Series B (Methodological) 46(2), 149–170 (1984)
17. Huber, P.J., et al.: Robust estimation of a location parameter. The Annals of Mathematical Statistics 35(1), 73–101 (1964)
18. Hunter, D.R., Lange, K.: A tutorial on mm algorithms. The American Statistician 58(1), 30–37 (2004)
19. Lange, K., Hunter, D.R., Yang, I.: Optimization transfer using surrogate objective functions. Journal of computational and graphical statistics 9(1), 1–20 (2000)
20. Le, H., Zach, C.: A graduated filter method for large scale robust estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5559–5568 (2020)
21. Mobahi, H., Fisher, J.W.: On the link between gaussian homotopy continuation and convex envelopes. In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 43–56. Springer (2015)
22. Mobahi, H., Zitnick, C.L., Ma, Y.: Seeing through the blur. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1736–1743. IEEE (2012)
23. Moré, J.J.: The levenberg-marquardt algorithm: implementation and theory. In: Numerical analysis, pp. 105–116. Springer (1978)
24. Neal, R.M., Hinton, G.E.: A view of the em algorithm that justifies incremental, sparse, and other variants. In: Learning in graphical models, pp. 355–368. Springer (1998)
25. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)
26. Parizi, S.N., He, K., Aghajani, R., Sclaroff, S., Felzenszwalb, P.: Generalized majorization-minimization. In: International Conference on Machine Learning, pp. 5022–5031. PMLR (2019)
27. Ribeiro, A.A., Karas, E.W., Gonzaga, C.C.: Global convergence of filter methods for nonlinear programming. SIAM Journal on Optimization 19(3), 1231–1249 (2008)
28. Rose, K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE 86(11), 2210–2239 (1998)
29. Sweeney, C.: Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org
30. Torr, P.H., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer vision and image understanding 78(1), 138–156 (2000)
31. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms, pp. 298–372. Springer (1999)
32. Wilson, K., Snively, N.: Robust global translations with 1dsfm. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
33. Wu, C.J.: On the convergence properties of the em algorithm. The Annals of statistics pp. 95–103 (1983)
34. Yang, H., Antonante, P., Tzoumas, V., Carbone, L.: Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. arXiv preprint arXiv:1909.08605 (2019)
35. Yuille, A.L., Rangarajan, A.: The concave-convex procedure. Neural computation 15(4), 915–936 (2003)
36. Zach, C.: Robust bundle adjustment revisited. In: European Conference on Computer Vision, pp. 772–787. Springer (2014)
37. Zach, C., Bourmaud, G.: Iterated lifting for robust cost optimization. In: British Machine Vision Conference (BMVC) (2017)