

Practical Methods For Convex Multi-View Reconstruction

Christopher Zach and Marc Pollefeys

ETH Zürich, Universitätstrasse 6, CH-8092 Zürich

Abstract. Globally optimal formulations of geometric computer vision problems comprise an exciting topic in multiple view geometry. These approaches are unaffected by the quality of a provided initial solution, can directly identify outliers in the given data, and provide a better theoretical understanding of geometric vision problems. The disadvantage of these methods are the substantial computational costs, which limit the tractable problem size significantly, and the tendency of reducing a particular geometric problem to one of the standard programs well-understood in convex optimization. We select a view on these geometric vision tasks inspired by recent progress made on other low-level vision problems using very simple (and easy to parallelize) methods. Our view also enables the utilization of geometrically more meaningful cost functions, which cannot be represented by one of the standard optimization problems. We also demonstrate in the numerical experiments, that our proposed method scales better with respect to the problem size than standard optimization codes.

1 Introduction

Globally optimal methods in multiple view geometry and 3D computer vision are appealing tools to estimate geometric entities from visual input. So far most research in this field has been focused on the formulation of geometric vision problems in terms of a standardized optimization taxonomy, e.g. as linear or higher order cone programs. With very few exceptions, generic optimization codes are utilized for the respective numerical experiments. The emphasis on global optimal formulations lies on L_∞ -based objective functions, i.e. minimizing the maximum over a set of convex or quasi-convex functions with respect to the unknowns. The initially intriguing L_∞ -based objective can be easily converted into a simple cost function combined with a large number of (convex) constraints, which subsequently enables tractable solvers to be applied. The decision of utilizing an L_∞ -based objective function has two important consequences: first, it induces particular (and often unrealistic) assumptions on the noise characteristic of the observed measurements; and second, the typically encountered quasi-convex nature of the optimization problem implies, that the solution procedure only indirectly provides the unknown variables of interest (through a sequence of feasibility problems). Hence, more robust and efficient alternative formulations of important tasks in multiple view geometry are desired.

Further, formulating geometric vision task in terms of general optimization problems has the advantage of having a well-understood theory and mature software implementations available, but such an approach also limits the range of multi-view problems and objective functions to those standard optimization problems. In this work we propose a more direct view on a class of geometric vision problems not taking the route through one of the standard convex programs. Our view on these problems is inspired by recent advances on convex formulations or relaxations of low-level vision problems. Our contribution is two-fold: we demonstrate the applicability of optimization methods primarily utilized in signal and image processing for geometric vision problems, and we extend recent convex models for multi-view reconstruction by a new cost function better approximating the squared reprojection error while still preserving the convexity of the problem.

2 Related Work

2.1 Global Optimization in Multiple View Geometry

Global optimization in multiple view geometry has gained a lot of interest in the recent years. In particular, L_∞ -based (or min-max) formulations are popular (i) due to the well-understood relation with fractional programming leading to linear or second order cone programs, and (ii) due to the good accuracy provided by the solution for geometric applications if no outliers are present.

The first exposition of L_∞ minimization for geometric computer vision problems is given in [1], where the authors propose the L_∞ cost function for multi-view reconstruction tasks. The relation between quasi-convex functions and L_∞ optimization for multi-view geometry was independently discovered in [2] and [3]. Quasi-convex functions (i.e. functions with convex sublevel sets) can be effectively minimized by a bisection search for the optimal value, thus solving a sequence of convex feasibility problems. Additional convex constraints can be also provided. These approaches present structure and motion recovery given known camera rotations as the prototypical application.

Sim and Hartley [4] discuss an L_∞ view on the problem of estimating camera centers (again under the assumption of known rotations) given the directions of the baseline between cameras. The 3D scene structure is not explicitly modeled as unknown parameter subject to minimization, therefore the problem size is substantially reduced. Placing camera centers given a set of relative directions is similar to the graph embedding problem for motion recovery [5, 6]. In [4] the degeneracy of embedding formulations for linear camera motions is addressed by utilizing the trifocal tensor to incorporate the relative scales of baselines. Removing the 3D structure from the problem formulation reduces the size of the optimization problem substantially, but also leads to minimization of quite abstract cost functions (e.g. the angular deviation between given and estimated baseline directions) instead of image-related quantities like the reprojection error.

The high computational costs of L_∞ optimization has lead to investigations to reduce the respective run-time. [7] describes an interior point algorithm ex-

exploiting the same sparsity pattern in the underlying problem as it is also found in sparse bundle adjustment methods. The observation that the objective value of a min-max problem is only dependent on a (potentially small) subset of error terms can be utilized to formulate faster methods for L_∞ optimization [8]. In this approach only a subset of data points is considered for optimization (thus making the problem smaller), but the residuals are evaluated for all data points. If all residuals are less or equal to the objective value, then the procedure can be stopped; otherwise additional data points with large residuals are added in further minimization steps.

L_∞ optimization has the potential disadvantage of being susceptible to outliers in the input data. It can be shown that some of the inputs attaining the maximum error in min-max optimization are guaranteed to be outliers under suitable assumptions, hence these outliers can be iteratively detected and removed by L_∞ optimization [9]. Alternatively, L_∞ (min-max) objective functions can be replaced by L_1 (min-sum) cost functions, leading directly to formulations much less affected by outliers in the data. Straightforward L_1 -based optimization of geometric vision problems similar to many of the L_∞ approaches outlined above usually leads to sum-of-fractions type of optimization problems, which are extremely difficult to solve. The recent approaches described in [10, 11] (and reviewed in more detail in Sec. 3) aim on directly minimizing the number (i.e. L_0 -norm) of outliers for a particular inlier criterion represented by suitable (either linear or second order cone) constraints. If all data points are inliers (i.e. the residuals are less than a given threshold), the objective value is 0. Convexification of the L_0 norm yields an L_1 -like objective function and thereby to respective linear or second order cone programs.

We present our method for practical convex optimization in multi-view geometry on the problem of structure and motion recovery given the global camera rotations. This raises the question of how these rotations can be determined. Global rotations can be efficiently computed from pair-wise relative orientations between images by utilizing the consistency relation between relative and absolute camera rotations, $R_j = R_{ij}R_i$. [12] and [13] present different methods to obtain consistent global rotations. For full structure and motion computation, [13] employs the L_∞ framework of [2], but uses only a small subset of scene points in order to accelerate the minimization procedure and to guarantee an outlier-free set of data points.

2.2 Non-smooth Convex Optimization and Proximal Methods

Even a convex optimization problem can be difficult to solve efficiently, especially if the objective is a non-smooth function. A class of methods successfully applied in signal and image processing are based on proximal calculus: for a convex function f and $\gamma > 0$ the mapping

$$\text{prox}_{\gamma f}(\bar{x}) = \arg \min_x \left\{ f(x) + \frac{1}{2\gamma} \|x - \bar{x}\|_2^2 \right\} \quad (1)$$

is called the proximity operator. It generalizes the notion of projection operators (for which f is the hard indicator function for a convex set S). In Eq. 1 the objective itself is called the Moreau envelope of function f of index γ . Sometimes $\text{prox}_{\gamma f}$ is difficult to compute directly, but the proximity operator for the conjugate function, $\text{prox}_{\gamma f^*}$, can be determined efficiently. In these cases we can utilize Moreau’s decomposition,

$$x = \text{prox}_{\gamma f}(x) + \gamma \text{prox}_{f^*/\gamma}(x/\gamma) \quad (2)$$

We refer to [14] for a recent, compact exposition of proximal calculus and its importance in image processing applications.

Proximal methods, in particular the forward-backward algorithm [14] and Douglas-Rachford splitting [15, 16], allow the efficient minimization of structured, convex problems of the form $\min_x f_1(x) + f_2(x)$. In brief, the Douglas-Rachford splitting iterates

$$\begin{aligned} \hat{x}^{(n)} &= \text{prox}_{\gamma f_2}(x^{(n)}) \\ x^{(n+1)} &= x^{(n)} + \text{prox}_{\gamma f_1}(2\hat{x}^{(n)} - x^{(n)}) - \hat{x}^{(n)}, \end{aligned} \quad (3)$$

and $\hat{x}^{(n)}$ is known to converge to the solution of $\min_x f_1(x) + f_2(x)$. See also e.g. [17] for the connections between Douglas-Rachford, augmented Lagrangian and split Bregman methods. In Section 4.3 we apply the Douglas-Rachford splitting on a problem with f_2 being the indicator function of a hyper-plane (i.e. $\text{prox}_{\gamma f_2}$ amounts to project the argument into a linear subspace), and f_1 further decomposing into many independent objectives.

3 Convex L_1 Reconstruction With Known Rotations

In this section we review robust structure and motion computation with known camera rotations based on convex optimization. Since all the camera centers and 3D points are mutually dependent in the objective function, we choose this application in order to demonstrate our method on a larger scale problem. Other classical problems addressed by global optimization in multi-view geometry are optimal point triangulation and camera resectioning, which involve a much smaller number of unknowns. In the following we assume that global camera rotations are given (e.g. a set of pairwise relative rotations can be upgraded to consistent global ones via eigenvalue decomposition [13]). Let i denote the index of the cameras and j be the index of 3D points, respectively. The set of global rotations R_i for each camera is assumed to be known. Further, let $u_{ij} = (u_{ij}^1, u_{ij}^2, 1)^T$ be the projection of the (unknown) 3D point X_j into image i , i.e. $u_{ij} \propto R_i X_j + T_i$, where T_i and X_j are the translation vectors and 3D point positions yet to be determined. We assume that the image coordinates u_{ij} are normalized, i.e. premultiplied by the inverse camera intrinsics. With known rotations, the relationship between 3D structure, camera poses and image measurements are essentially linear (up to the unknown depth). The full projection

function

$$\hat{u}_{ij} = \frac{R_i X_j + T_i}{(R_i X_j + T_i)_3}$$

is nonlinear, but e.g. the squared reprojection error is quasi-convex and amenable for L_∞ optimization (e.g. [2]). We focus on the L_1 setting where a minimizer of the sum of some deviation is sought. The intention is to increase robustness in presence of gross outliers by utilizing an L_1 objective function. [10, 11] use a quasi L_∞ model by assigning zero cost, whenever the projection of a 3D point X_j lies within a neighborhood of a user-specified radius σ (where the neighborhood is induced either by the Euclidean norm [11] or by the maximum norm [10]). Consequently, no cost is attributed in the objective function, if X_j lies within a (generalized) cone induced by the camera center $C_i = -R_i^T T_i$ and the observed image point u_{ij} (see Figure 1).

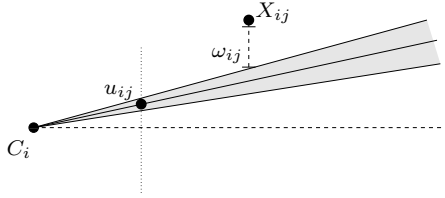


Fig. 1. The cone induced by the camera center C_i and the observed image point u_{ij} . Points X_{ij} residing within the shaded cone are considered as inliers and have no cost in the objective function, whereas outliers are penalized.

Denoting $X_{ij} = (X_{ij}^1, X_{ij}^2, X_{ij}^3)^T = R_i X_j + T_i$, the condition of X_j being in the respective cone with radius σ reads as

$$\|u_{ij} - X_{ij}^{1,2}/X_{ij}^3\|_p \leq \sigma$$

or equivalently,

$$\|u_{ij} X_{ij}^3 - X_{ij}^{1,2}\|_p \leq \sigma X_{ij}^3, \quad (4)$$

where we also employ the chirality constraint of X_j being in front of the camera i , i.e. $X_{ij}^3 \geq 0$. The underlying norm can be the L_1 norm ($p = 1$), the Euclidean one ($p = 2$), or the maximum norm ($p = \infty$). Observe that the constraint Eq. 4 corresponds to a second order cone ($p = 2$) and the intersection of affine linear half-spaces, respectively.

If T_i 's and X_j 's can be determined such that Eq. 4 is fulfilled for all i and j , then all reprojection errors are less or equal to σ (either using L_2 or L_∞ distances in the image). If this is not the case, one can measure the infeasibility s_{ij} of the projected 3D point [11],

$$\|u_{ij} X_{ij}^3 - X_{ij}^{1,2}\|_p \leq \sigma X_{ij}^3 + s_{ij},$$

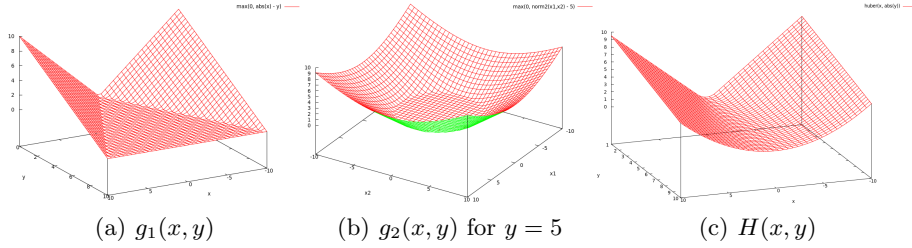


Fig. 2. Surface plots for the two pan functions g_1 (a) and g_2 (for a fixed value of y , (b)), and the bivariate Huber cost function (c).

or the necessary offset vectors in object space to move X_{ij} onto the cone, $\omega_{ij} \in \mathbb{R}^2$ [10],

$$\|u_{ij}X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij}\|_p \leq \sigma X_{ij}^3.$$

Since nonzero values of s_{ij} or ω_{ij} correspond to outlier measurements in the image, it is reasonable to search for sparse solutions in terms of s_{ij} and ω_{ij} , respectively, i.e. to minimize the L_0 norm of s_{ij} (or ω_{ij}). Convexification of the L_0 norm yields the following L_1 objective function and constraints (using the offset vector formulation):

$$\begin{aligned} \min_{T_i, X_j, \omega_{ij}} \sum_{ij} \|\omega_{ij}\|_1 \quad \text{s.t.} \\ \|u_{ij}X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij}\|_p \leq \sigma X_{ij}^3 \quad \forall ij \\ X_{ij} = R_i X_j + T_i. \end{aligned} \quad (5)$$

If $p = \infty$ (and also for $p = 1$) this is a linear program, and for $p = 2$ one obtains a second order cone program, which can be solved by suitable convex optimization codes.

In order to avoid the degenerate solution $T_i = 0$ and $X_j = 0$ for all cameras and 3D points in the optimization problem Eq. 5 and to avoid a 4-parameter family of solutions (arbitrary global translation and scale), one has to enforce suitable cheirality constraints (e.g. $X_{ij}^3 \geq 1$ [10]) or fix the reference frame [11]. Utilizing the cheirality constraint implicitly selects the smallest feasible reconstruction with respect to its scale, since the objective function in Eq. 5 is reduced by decreasing the global scale.

4 Our Approach

Generic optimization of Eq. 5 using a linear or second order cone programming toolbox turns out to be not efficient in practice. One reason for the inefficiency is the introduction of auxiliary variables, either s_{ij} or ω_{ij} . Another difficulty for

generic optimization codes is the large number of non-local constraints. Hence, we propose to directly optimize a non-differentiable objective function without the need for additional unknowns.

4.1 The Cost Functions

In order to reformulate Eq. 5 in more general terms, we define and analyze several convex functions, which are used in Sec. 4.2 to derive suitable proximal-point problems forming the basis of the numerical scheme.

Notations We introduce the indicator function $\iota_S(x)$ returning 0 if $x \in S$ and ∞ otherwise. In particular, $\iota_{\mathbb{R}_0^+}$ and $\iota_{\mathbb{R}_0^-}$ denote the indicator functions for non-negative (respectively non-positive) real numbers. For a convex, lower semi-continuous function f , let f^* be its conjugate function, $f^*(z) = \max_x zx - f(x)$.

The ‘‘Pan’’ Functions We define the pan function $g_d : \mathbb{R}^d \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ as

$$g_d(x, y) = \max \{0, \|x\|_2 - y\}. \quad (6)$$

For a particular value of $y \geq 0$ the shape of $g_d(\cdot, y)$ is a truncated L_1 cost function resembling the cross-section of a pan-like shape (see also Fig. 2(a) and (b)). It can be also viewed as two-sided variant of a hinge loss function. This function is convex, but not differentiable at $\|x\|_2 = y$.

As a first step to derive the conjugate function of g_d , we observe that

$$g_d(x, y) = \max_{\|z\|_2 \leq 1} \{z^T x - \|z\|_2 y\}. \quad (7)$$

Omitting the subscript in $\|\cdot\|_2 = \|\cdot\|$, if $\|x\| \leq y$ we have $z^T x - \|z\|y \leq \|z\| \|x\| - \|z\|y = \|z\|(\|x\| - y)$ (by the Cauchy-Schwarz inequality), but the second factor is non-positive by assumption, hence maximizing over z in the unit disc yields $z = 0$ with objective value 0. $\|x\| > y$: observe that $\|z\|y$ is independent of the direction of z and $z^T x$ is maximal if $z \parallel x$, i.e. $z = kx$ for some $k \in [0, \|x\|^{-1}]$ (since $\|z\| \leq 1$). Overall, $z^T x - \|z\|y = k\|x\|^2 - k\|x\|y = k\|x\|(\|x\| - y)$, hence $k = \|x\|^{-1}$ (i.e. $z = x/\|x\|$) maximizes that expression with value $\|x\| - y$. Overall, both definitions Eqs. 6 and 7 are again equivalent.

Finally, we can convert Eq. 7 into a bilinear form by introducing the additional variable v ,

$$g_d(x, y) = \max_{\|z\| \leq 1, v \leq -\|z\|} \{z^T x + vy\}. \quad (8)$$

Note that $g_d(x, y)$ is ∞ for $y < 0$, since v is not bounded from below. For given $x \in \mathbb{R}^d$ and $y > 0$ the maximization always gives $v = -\|z\|$, since the objective can be increased whenever $v < -\|z\|$. Thus, the definitions in Eqs. 7 and 8 are equal, and Eq. 8 allows us to directly read off the corresponding conjugate function $g_d^*(z, v) = \iota_{C_d}(z, v)$ with $C_d \equiv \{(z, v) : \|z\| \leq 1, \|z\| \leq -v\}$. Thus, we can reduce the computation of $\text{prox}_{\gamma g_d}$ essentially to the projection into the set C_d in view of Moreau’s decomposition Eq. 2. Projecting into C_d can be done in closed form and distinction of cases.

The Bivariate Huber Cost Function Instead of having a combined L_∞/L_1 cost function as described in the previous section, one can also consider penalizing a squared residual for inliers as defined by the respective 3D cone, and an L_1 penalizer for outliers. Define the bivariate Huber cost function $H : \mathbb{R} \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ by

$$H(x, y) = \begin{cases} \frac{x^2}{2y} & |x| \leq y \\ |x| - y/2 & |x| \geq y \end{cases} \quad (9)$$

for $y \geq 0$ (see also Fig. 2(a)). We denote this function as bivariate Huber cost since it also takes the inlier threshold (here y) as additional parameter. Unlike the squared reprojection error, which is only a quasi-convex function, $H(x, y)$ is convex in $\mathbb{R} \times \mathbb{R}_0^+$. The conjugate function of the univariate Huber cost (i.e. as a function only of x) is readily derived as

$$H^*(z; y) = \frac{y}{2}z^2 + \iota_{[-1,1]}(z). \quad (10)$$

Note that partial conjugation with respect to x is not sufficient for our purpose, since $\frac{y}{2}z^2$ is not a bilinear expression in the primal and dual variables. We combine $H^*(z; y)$ with $\iota_{\mathbb{R}_0^+}(y)$ corresponding to the constraint $y \geq 0$, and obtain

$$\begin{aligned} H(x, y) + \iota_{\mathbb{R}_0^+}(y) &= \max_{z \in [-1,1]} \left(zx - \frac{y}{2}z^2 \right) + \max_{v \leq 0} vy \\ &= \max_{\substack{z \in [-1,1] \\ v' \leq -z^2/2}} zx + v'y, \end{aligned}$$

where we substituted $v' = v - z^2/2$ and used the fact $\iota_{\mathbb{R}_0^+}(y) = \max_{v \leq 0} vy$. Note that the last line allows us to immediately identify the conjugate function of H with respect to both arguments as

$$H^*(z, v) = \iota_{[-1,1]}(z) + \iota_{\mathbb{R}_0^-}(v + z^2/2). \quad (11)$$

The feasible region for (z, v) is the region K below the parabola $-z^2/2$ intersected by $[-1, 1] \times \mathbb{R}$. Finding the closest point in K requires to determine the nearest point on a parabola segment, which leads to solving a 3rd order polynomial. Luckily, this cubic polynomial is in depressed form (i.e. of the form $x^3 + px + q = 0$) and is known to have only one real root. Hence, projecting an arbitrary pair (z, v) into the set K is tractable although not extremely cheap (due to the necessary computation of cubic roots).

The cost functions and the respective conjugates introduced in this section allow the efficient application of proximal methods to the multi-view reconstruction problem as discussed in the following.

4.2 Application to Multi-View Reconstruction

Our formulation of the multi-view reconstruction approach with known camera rotations follows very closely the model Eq. 5. In our experiments we observed

that solely fixing the gauge freedom by setting two translation vectors can still result in quasi-degenerate solutions. The utilized L_1 penalizer on the offset variables measures deviations in object space (in contrast to image space) and therefore induces a strong prior towards solutions collapsing many of the scene points and camera centers into a single point. Further, strictly enforcing the cheirality constraint for all 3D points is very restrictive and does not cope well with outlier correspondences ultimately triangulated behind one of the cameras. Therefore, we slightly modify the convex formulation of Eq. 5 into

$$\begin{aligned} \min_{T_i, X_j, \omega_{ij}, \rho_{ij}} \sum_{ij} \|\omega_{ij}\|_1 + \sum_{ij} [\rho_{ij}]_+ \quad \text{s.t.} \quad (12) \\ \|u_{ij}^{1,2} X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij}^{1,2}\| \leq \sigma(X_{ij}^3 + \rho_{ij}) \\ X_{ij}^3 + \rho_{ij} \geq 1 \quad X_{ij} = R_i X_j + T_i, \end{aligned}$$

where $[\cdot]_+ \equiv \max\{0, \cdot\}$. Hence, we look for sparse correction values ω_{ij} and ρ_{ij} such that all corrected 3D points lie inside the cone induced by the measured image projection u_{ij} (first inequality constraint) and fulfill the cheirality constraint (second inequality constraint). By observing the following equivalence,

$$g_d(x, y) = \min_{s \in \mathbb{R}^d: \|x+s\|_2 \leq y} \|s\|_2,$$

we can use the pan function introduced in Sec. 4.1 to eliminate the auxiliary variables ω_{ij} , and obtain the equivalent problem (illustrated first for $d = 1$, i.e. the anisotropic variant of the inlier cone):

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} \sum_{l=1}^2 g_1(u_{ij}^l \bar{X}_{ij}^3 - X_{ij}^l, \sigma \bar{X}_{ij}^3) + \sum_{ij} \iota_{[1, \infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+, \quad (13)$$

with $X_{ij} = R_i X_j + T_i$ and $\bar{X}_{ij}^3 = X_{ij}^3 + \rho_{ij}$. The choice $d = 2$ leads to a similar (now isotropic) problem using g_2 instead of g_1 :

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} g_2(u_{ij}^{1,2} \bar{X}_{ij}^3 - X_{ij}^{1,2}, \sigma \bar{X}_{ij}^3) + \sum_{ij} \iota_{[1, \infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+. \quad (14)$$

Both problems are convex minimization tasks with non-differentiable objective functions. Observe that the arguments of g_d and ι are linear in the unknowns T_i , X_j and ω_{ij} .

Finally, instead of having just zero cost for inliers (as in the two objectives above), squared deviations from the observed image points over the distance can be penalized by utilizing the bivariate Huber function,

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} \sum_{l=1}^2 H(u_{ij}^l \bar{X}_{ij}^3 - X_{ij}^l, \sigma \bar{X}_{ij}^3) + \sum_{ij} \iota_{[1, \infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+. \quad (15)$$

This formulation essentially approximates the squared reprojection error (i.e. squared numerator and denominator) by a convex quadratic-over-linear function

for inlier points. Consequently, the 3D points in the corresponding solution tend to stay closer to the observed image measurements. More importantly, inlier 3D points are attracted by a unique minimum and the numerical procedure converges faster in practice.

At the first glance nothing is gained by such reformulation other than moving the (linear or second order cone) constraints into the objective (Eqs. 13 and 14), and allowing for a refined cost for inlier points (Eq. 15). But the problems are very structured: the objective function is a sum of many convex functions only taking very few arguments, therefore depending only on a small subset of the unknowns. Hence, a variant of Douglas-Rachford splitting can be applied as described in the following section.

4.3 Numerical Scheme

The objective functions in the previous section can be more generally written as

$$\min_{\mathcal{X}} \sum_k h_k(L_k \mathcal{X}), \quad (16)$$

where \mathcal{X} denotes all unknowns T_i , X_j and ρ_{ij} , the h_k are convex functions and L_k are matrices of appropriate dimensions. Similar to dual decomposition methods we can introduce local unknowns x_k for each h_k and explicitly enforce global consistency (see also [18]),

$$\min_{\mathcal{X}, \mathcal{Y}_k} \underbrace{\sum_k h_k(\mathcal{Y}_k)}_{\equiv f_1} + \underbrace{\sum_k \iota\{L_k \mathcal{X} = \mathcal{Y}_k\}}_{\equiv f_2}. \quad (17)$$

Application of Douglas-Rachford splitting amounts to solving $\text{prox}_{\gamma f_1}$ and $\text{prox}_{\gamma f_2}$ (recall Eq. 3). The first proximity operator, $\text{prox}_{\gamma f_1}$ decouples into independent problems $\text{prox}_{\gamma h_k}$, in particular (referring to Eq. 15, with analogous expressions for Eqs. 13 and 14) the term $h_k(L_i \mathcal{X})$ is one of

$$h_k(L_i \mathcal{X}) = \begin{cases} H(u_{ij}^l \bar{X}_{ij}^3 - X_{ij}^l, \sigma \bar{X}_{ij}^3) & l \in \{1, 2\} \\ \iota_{[1, \infty)}(\bar{X}_{ij}^3) \\ [\rho_{ij}]_+ \end{cases}$$

For h_k equal to g_d or H we can utilize the derivations from Section 4.1 in order to determine $\text{prox}_{\gamma g_d}$ or $\text{prox}_{\gamma H}$ efficiently. If $h_k = \iota_{[1, \infty)}(\cdot)$, the proximity operator can be easily derived as clamping operation into the feasible domain $[1, \infty)$, and for $h_k = [\cdot]_+$ the respective proximity operator is given by

$$\text{prox}_{\gamma[\cdot]_+}(\bar{x}) = \max(0, \bar{x} - \gamma).$$

$\text{prox}_{\gamma f_2}(\mathcal{X}, (\mathcal{Y}_k)_k)$ corresponds to finding the closest point $(\hat{\mathcal{X}}, (\hat{\mathcal{Y}}_k)_k)$ satisfying the linear constraints $L_k \hat{\mathcal{X}} = \hat{\mathcal{Y}}_k$ for all k , therefore $\text{prox}_{\gamma f_2}$ is a projection operation into a linear subspace. Following [19] a particular instance of a Douglas-Rachford approach called simultaneous-direction method of multipliers (SDMM)

can be stated: choose $\gamma > 0$ and arbitrary initial values $y_k^{(0)}$ and $z_k^{(0)}$, and iterate for $n = 0, 1, \dots$:

$$\begin{aligned} x^{(n)} &= \left(\sum_k L_k^T L_k \right)^{-1} \sum_k L_k^T \left(y_k^{(n)} - z_k^{(n)} \right) \\ y_k^{(n+1)} &= \text{prox}_{\gamma h_k} \left(L_k x^{(n)} + z_k^{(n)} \right) \quad \forall k \\ z_k^{(n+1)} &= z_k^{(n)} + L_k x^{(n)} - y_k^{(n+1)} \quad \forall k. \end{aligned} \tag{18}$$

$x^{(n)}$ converges to a solution of Eq. 16. Note that the L_k are very sparse matrices, and the inverse of $\sum L_k^T L_k$ can be efficiently found using sparse Cholesky factorization. Note that a projection into a linear subspace is always uniquely defined, hence $\sum L_k^T L_k$ must have full rank.

Since the method is iterative, a suitable stopping criterion is required. We selected the following one: the current 3D structure and the one obtained after a significant number of iterations (our choice is 1000) are normalized in their sizes, and the maximum change in the respective 3D point positions is used as stopping criterion. If no 3D point was moved more than ε , then the update iterations terminate. We use $\varepsilon = 10^{-8}$ in our implementation. Further, we set $\gamma = 0.1$ in all experiments.

5 Numerical Results

Most work in convex and quasi-convex optimization in multiple view geometry uses the ‘‘Dinosaur’’ data-set¹ consisting of camera poses and tracked 2D points. We use the rotations known from the given projection matrices, and determine the 3D structure and camera centers using the convex formulations described in Sec. 4.2. We obtain numerical and timing results for several optimization codes: (i) we utilize an easy-to-use simplex-based linear program solver² to optimize the problem Eq. 5 (denoted as ‘‘Simplex’’ in the evaluation Table 1); (ii) we also experiment with an interior point algorithm for semi-definite programs (DSDP [20], which has also a direct interface to specify LP cones, indicated by DSDP in Table 1)); and finally we implemented the update equations Eq. 18 for the described cost functions to minimize the respective energy. Table 1 summarizes the obtained timing and accuracy results, where we set the inlier radius σ to one pixel. Of particular interest in the evaluation is the dependence of the runtime on the data-set size: generally, the iterative proximal methods scale better with the data-set size. Both generic LP solvers were not able to complete the full data-set (due to numerical instabilities for the Simplex method and excessive memory consumption of the semi-definite code DSDP).

These performance numbers need to be compared with the close to two hours reported in [11] for the full data-set. Note that [10] does not indicate the respective run-time performance. The Huber function based model Eq 15 is strictly

¹ available from <http://www.robots.ox.ac.uk/~vgg/data/data-mview.html>

² <http://lpsolve.sourceforge.net/>

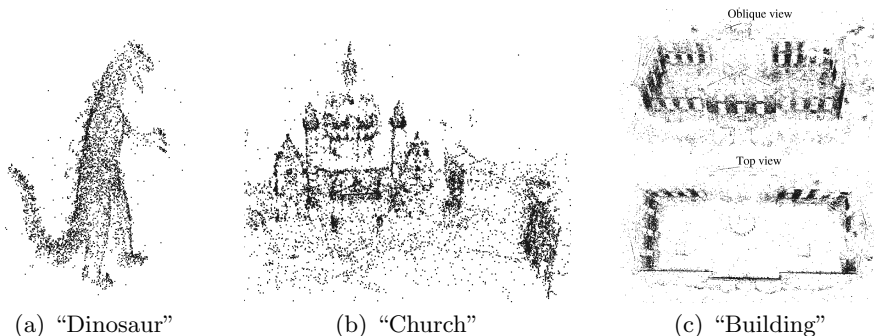


Fig. 3. Reconstruction result for the “Dinosaur” data set (36 images, ≈ 5.000 3D points), “Church” data set (58 views, ≈ 12.500 3D points), and “Building” (128 images, ≈ 27.000 3D points) using the energy function Eq 15. No bundle adjustment is applied.

convex for inlier points (and therefore has a unique minimizer), which could be the explanation for reaching the stopping criterion faster than the other energy models Eqs. 13 and 14.

Since the Huber cost model Eq. 15 penalizes deviations from the ray induced by the image measurements, the final reprojection error of the 3D points are consistently smaller than for the combined L_∞/L_1 cost models. The last column in Table 1 depicts the number of reported outlier measurements outside the respective σ radius in the image. This value is stated to show the equivalence of the iterative method for Eq. 13 with the linear programming formulation. Further, it shows that the smaller reprojection error in the Huber cost model is compensated by a larger number of reported outliers (which is a result of the energy model and not induced by the implementation). We also applied the proposed method on real-world data sets consisting of 58 and 128 views, respectively (see Fig. 3(b) and (c)). Consistent rotations are estimated via relative poses from SIFT feature matches. The iterative Huber approach (for $\sigma = 2$ pixels) requires 12.5 and 40 minutes to satisfy the convergence test, and about 30% is spent in approximate column reordering for the sparse Cholesky factorization. The mean reprojection errors for inliers are 1.08 and 0.53 pixels, respectively. Figure 4 depicts the convergence rate of the objective value for the Dinosaur data set with respect to the number of iterations. Since the objective function is rather flat near the global minimum, small changes in the objective value do not necessarily imply small updates in the variables, and the termination criterion is achieved much later.

6 Conclusion

In this work we present a different view on convex problems arising in multiple-view geometry, and propose a non-standard optimization method for these problems. By looking at classical optimization tasks in geometric vision from a general

# views	# 3D points	Method	Run-time	L_∞ inlier error	L_2 inlier error	# of outliers
8	843	Simplex	9.5s	0.913	1.17	33/2465
		DSDP	2m27s	0.430	0.480	33/2465
		Anisotr.	11.4s	0.622	0.712	33/2465
		Isotropic	4s	0.466	0.511	32/2465
		Huber cost	5.4s	0.223	0.235	48/2465
16	1915	Simplex	1m43s	0.892	1.10	165/5905
		DSDP	24m37s	0.389	0.434	165/5905
		Anisotr.	26.1s	0.705	0.800	165/5905
		Isotropic	13.5s	0.668	0.721	165/5905
		Huber cost	13.7s	0.256	0.271	207/5905
24	3205	Simplex	19m26s	0.895	1.10	294/10416
		DSDP	103m36s	0.369	0.411	294/10416
		Anisotr.	1m44s	0.895	0.756	294/10416
		Isotropic	1m33s	0.784	0.838	311/10416
		Huber cost	1m16s	0.262	0.278	371/10416
36	4983	Anisotr.	1m51s	0.650	0.734	383/16432
		Isotropic	45s	0.483	0.533	421/16432
		Huber cost	46s	0.269	0.286	539/16432

Table 1. Accuracy and run-time results for the “Dinosaur” sequence for increasing data-set sizes. Results are shown for Simplex and interior-point based (DSDP) solvers optimizing the model Eq. 5, and proposed iterative methods for all energy models.

convex optimization perspective we can formulate interesting new geometric cost functions and also provide practical minimization procedures.

Future work needs to explore other applications in geometric computer vision potentially taking advantage of the proposed formulation and the associated numerical method, e.g. for quasi-convex problems. Finally, GPU-based implementations of the numerical methods are expected to result in significant reductions of the run-time for the proposed methods.

Acknowledgements: We would like to thank Manfred Klopschitz and Alexander Schwing for their valuable support and feedback.

References

1. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: Proc. CVPR. (2004)
2. Kahl, F.: Multiple view geometry and the L_∞ norm. In: Proc. ICCV. (2005) 1002–1009
3. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: Proc. ICCV. (2005) 986–993
4. Sim, K., Hartley, R.: Recovering camera motion using L_∞ minimization. In: Proc. CVPR. (2006)

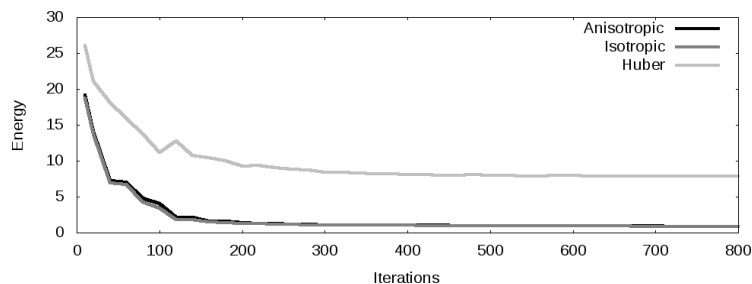


Fig. 4. Temporal evolution of the objective value with respect to iterations for the full Dinosaur dataset. 100 iterations corresponds to 0.6s (anisotropic and isotropic objectives) and 1.1s (Huber cost function) of run-time.

5. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proc. CVPR. (2001) 218–225
6. Brand, M., Antone, M., Teller, S.: Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In: Proc. ECCV. (2004)
7. Seo, Y., Lee, H.L.S.W.: Sparse structures in L-Infinity norm minimization for structure and motion reconstruction. In: Proc. ECCV. (2006) 780–793
8. Seo, Y., Hartley, R.: A fast method to minimize L_∞ error norm for geometric vision problems. In: Proc. CVPR. (2007)
9. Sim, K., Hartley, R.: Removing outliers using the L_∞ norm. In: Proc. CVPR. (2006)
10. Dalalyan, A., Keriven, R.: L_1 -penalized robust estimation for a class of inverse problems arising in multiview geometry. In: NIPS. (2009)
11. Seo, Y., Lee, H.L.S.W.: Outlier removal by convex optimization for l-infinity approaches. In: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology. (2009) 203–214
12. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proc. CVPR. (2004) 684–691
13. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Proc. CVPR. (2007)
14. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation* **4** (2005) 1168–1200
15. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16** (1978) 964–979
16. Combettes, P.L., Pesquet, J.C.: A douglas-rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Selected Topics Signal Processing* **1** (2007) 564–574
17. Esser, E.: Applications of lagrangian-based alternating direction methods and connections to split bregman. *UCLA CAM Report TR09-31* (2009)
18. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. *Journal of Visual Communication and Image Representation* (2009)
19. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing (2009) arXiv:0912.3522v2.
20. Benson, S.J., Ye, Y., Zhang, X.: Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization* **10** (2000) 443–461