

Outlier Removal by Convex Optimization for L-Infinity Approaches

Yongduek Seo, Hyunjung Lee, and Sang Wook Lee

Department of Media Technology, Sogang University, Korea
{yndk, whitetobi, slee}@sogang.ac.kr

Abstract. This paper is about removing outliers without iterations in L_∞ optimization. Existing L_∞ outlier removal method requires iterative removal of the set of measurements with greatest residual during L_∞ minimization. In the method presented in this paper, on the other hand, a threshold is preset once for the maximum residual error in a manner similar to RANSAC, and the measurements yielding greater residuals than the threshold are taken to be outliers. We examine two feasibility test algorithms: 1) one that minimizes the maximum infeasibility and 2) the other that minimizes the sum of infeasibilities (SOI). Both of these can be used for feasibility test in conjunction with the bisection algorithm which attains the L_∞ optimum. We note that the SOI method has an interesting characteristic due to its L1-norm minimization nature. It tries to estimate a robust solution while maximizing the number of feasible constraints. The infeasible constraints are found to be due mostly to outliers. Once we set a threshold, the SOI algorithm sorts out outliers from the data set without any repetition and substantial reduction of computation time can be achieved compared to the iterative method. Experiments with synthetic as well as real objects demonstrate the effectiveness of the SOI method. We suggest that the SOI method precede the outlier-sensitive L_∞ optimization.

1 Introduction

The L_∞ -norm minimization has recently received a great deal of attention in the area of geometric vision since it yields global optimum because of the quasi-convexity of the L2 re-projection error function [1,2,3]

It can be used to find the global optimum for the problems such as triangulation, inter-image homography, camera resectioning, structure and motion with known rotation or homography. Research topics related to L_∞ optimization include motion estimation [4], robot application [5], non-rigid surface tracking [6], outliers removal [7,8], increasing the speed of computation [9,10], and the pseudo-convexity of the re-projection error function [9].

Removing outliers. A major weakness of the L_∞ optimization lies in its susceptibility to outliers since the minimization actually fits the outliers instead of the good data. RANSAC [11] removes most of the outliers during the first stage of computation, but there is still a possibility that some of the outliers remain. Fortunately, one can easily eliminate those remaining outliers by iteratively optimizing and removing the measurements with greatest residual. Sim and Hartley showed that this iterative outlier

elimination is valid for a wide class of L_∞ problems by proving that the set of measurements with greatest residual, called the support set, must contain at least one outlier [7]. This iterative scheme has been employed successfully in [6] for 3D deformable surface tracking. The most notable disadvantage of the iterative outlier removal is the long computation time, especially for a problem with a large number of parameters such as the estimation of structure and motion. For such a problem, the computation can easily take much longer than 10 hours with a typical high-performance PC. It may be noted that Li developed a method of removing only outliers rather than the whole support set [8]. Despite its improved efficiency, however, the method still requires a number of tests and considers only the triangulation problem. No framework was provided for applying the method to other problems with larger number of parameters such as homography computation, and motion and structure estimation with known rotation.

Goal of the presented work. This paper challenges the problem of computationally efficient outlier removal and suggests an algorithm that does not require iterations for removing outliers. Presented is a one-shot outlier-removal method. We simply specify a maximum threshold for re-projection errors, run a feasibility test algorithm, and regard the infeasible measurements as outliers. Then the L_∞ optimization is carried out once only on the measurements identified as inliers. We experimentally show that this one-shot method reduces the computation time by a factor of $10 \sim 133$ for some typical L_∞ problems compared to the iterative method [7]. The feasibility test is a robust convex optimization algorithm with the second order cone constraints. Like [7], we advocate the use of RANSAC at the initial stage of outlier removal. Our method is intended to get rid of the outliers that may remain even after RANSAC is performed.

Feasibility problem. L_∞ optimization is done usually by the bisection method shown in Algorithm 1. Recent study [9] on the pseudo-convexity of the error function opens up a possibility of a new way of fast computation without resorting to the bisection method although a concrete numerical algorithm is yet to be developed for efficient computation. Most of the works on L_∞ minimization heavily rely on the use of a readily available convex solver such as SeDuMi implemented in Matlab [12]. The simplex algorithm can be used for the methods which employ linear programming rather than second-order cone programming (SOCP) [13,10]. In most of these works, the feasibility test problem in Algorithm 1 is not explicitly described. We focus on the feasibility test formulation to find a way to identify outliers. The feasibility problem in Algorithm 1 may be solved numerically by an SOCP formulation which checks the feasibility of the convex constraints and finds a solution. We consider two of such formulations: 1) minimizing the maximum infeasibility and 2) minimizing the sum of the infeasibilities (SOI). The first formulation is appropriate for testing the feasibility of the whole set of constraints. The second allows us to see which are feasible among the constraints and to identify inliers. We call this the SOI method on which our one-shot non-iterative outlier-removal algorithm is based.

Section 2 discusses those two formulations in detail after providing a brief introduction to L_∞ error norm minimization, and our strategy for outlier removal is described. Section 3 shows various experiments with synthetic data to demonstrate the performance of the SOI method for camera resectioning problem. Section 4 presents

Algorithm 1. Bisection method to minimize L_∞ norm

Input: initial upper(U)/lower(L) bounds, tolerance $\epsilon > 0$.

```

repeat
     $\gamma := (L + U)/2$ 
    Solve the feasibility problem (7)
    if feasible then  $U := \gamma$  else  $L := \gamma$ 
until  $U - L \leq \epsilon$ 
    
```

our experimental results for the problem of structure and motion with known rotation. Experiments with real data, the well-known dinosaur sequence, are carried out as well as with a set of synthetic data. Concluding remarks are given in Section 5.

2 The L_∞ Optimization and Feasibility Test

We take the resectioning problem as a specific example of L_∞ optimization for explaining our approach. Other problems can be formulated in a similar way based on the work shown in [2]. A good introduction to the feasibility problems can be found in [14].

Given a set of N correspondences $\{\mathbf{X}_i, \mathbf{x}_i\}$ of a 3D point $\mathbf{X}_i = [X_i, Y_i, Z_i, 1]^\top$ and its noisy image measurement $\mathbf{x}_i = [u_i, v_i]^\top$, the resectioning problem is to find the camera matrix \mathbf{P} such that

$$[u_i, v_i] = \left[\frac{\mathbf{p}_1 \mathbf{X}_i}{\mathbf{p}_3 \mathbf{X}_i}, \frac{\mathbf{p}_2 \mathbf{X}_i}{\mathbf{p}_3 \mathbf{X}_i} \right], \quad \forall i \in I = \{1, \dots, N\}, \tag{1}$$

where \mathbf{p}_k is the k th row vector of the 3×4 matrix \mathbf{P} . The L_2 reprojection error is defined to be

$$e_i = \sqrt{\left(u_i - \frac{\mathbf{p}_1 \mathbf{X}_i}{\mathbf{p}_3 \mathbf{X}_i}\right)^2 + \left(v_i - \frac{\mathbf{p}_2 \mathbf{X}_i}{\mathbf{p}_3 \mathbf{X}_i}\right)^2} \tag{2}$$

$$= \frac{\| [u_i \mathbf{p}_3 \mathbf{X}_i - \mathbf{p}_1 \mathbf{X}_i, \quad v_i \mathbf{p}_3 \mathbf{X}_i - \mathbf{p}_2 \mathbf{X}_i] \|_2}{\mathbf{p}_3 \mathbf{X}_i} \tag{3}$$

$$= \frac{\| \mathbf{A}_i \boldsymbol{\theta} + \mathbf{b}_i \|_2}{\mathbf{c}_i^\top \boldsymbol{\theta} + d_i}, \tag{4}$$

where $\| \cdot \|_2$ is the Euclidean norm, $\boldsymbol{\theta}$ is the vector form of \mathbf{P} , and $\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i$ are appropriate coefficient matrices, vectors and a scalar. The total error is $\mathbf{E} = [e_1, \dots, e_N]$ and the optimum of the reprojection errors in the L_∞ error sense is given by

$$e_\infty = \min_{\boldsymbol{\theta}} \max\{e_1(\boldsymbol{\theta}), \dots, e_N(\boldsymbol{\theta})\}. \tag{5}$$

Note that finding the optimum can be re-written as

$$\begin{aligned} \min \quad & \gamma \\ \text{s. t.} \quad & \| \mathbf{A}_i \boldsymbol{\theta} + \mathbf{b}_i \| \leq \gamma (\mathbf{c}_i^\top \boldsymbol{\theta} + d_i) \quad \forall i \in I. \end{aligned} \tag{6}$$

Since the error function in Equation 4 is quasi-convex, the constraints are all convex second order cones for a fixed constant γ . Therefore, the optimum and the solution θ^* can be found by the bisection method shown in Algorithm 1, where the following feasibility problem with the constant γ should be solved at every iteration:

Problem 1

$$\begin{aligned} & \text{find } \theta \\ & \text{s.t. } \|A_i\theta + b_i\| \leq \gamma(c_i^\top \theta + d_i) \quad \forall i \in I. \end{aligned} \tag{7}$$

2.1 Minimizing the Maximum Infeasibility

Note that the description of Problem 1 neither explicitly specifies how to test the feasibility, nor shows how to find a feasible solution θ numerically inside the second order cones even when a good convex optimizer is available. A numerically solvable description of the problem can be obtained by introducing a new auxiliary variable s as:

$$\begin{aligned} & \min s \\ & \text{s.t. } \|A_i\theta + b_i\| \leq \gamma(c_i^\top \theta + d_i) + s \quad \forall i \in I. \end{aligned} \tag{8}$$

By setting $\tilde{\theta} = [\theta; s]$, one may easily see that this formulation is exactly in the form of second order cone programming:

$$\begin{aligned} & \min f_0^\top \tilde{\theta} \\ & \text{s.t. } \|\tilde{A}_i \tilde{\theta} + b_i\| \leq (\tilde{c}_i^\top \tilde{\theta} + \gamma d_i) \quad \forall i \in I, \end{aligned} \tag{9}$$

where $f_0^\top = [0; 1]$, and $\tilde{A}_i = [A_i | 0]$ and $\tilde{c}_i = [\gamma c_i^\top, 1]^\top$ are formed by augmenting A_i and γc_i with a column of zeros and a row of one, respectively. The variable s represents the maximum infeasibility for the given constraints, i.e., the whole set of the second order cones. When $s > 0$, the problem is infeasible; otherwise, it is feasible. L_∞ norm minimization can be carried out by minimizing s . This minimization based on Equation 9 needs only one extra variable s , and thus is simpler to implement than the following alternative feasibility test.

2.2 Minimizing the Sum of Infeasibilities (SOI)

To minimize SOI, we use as many variables s_i s ($i = 1, \dots, N$) as the number of measurements (or equivalently second order cones), i.e., we have one s_i per one measurement. The problem description is given as:

$$\begin{aligned} & \min s_1 + s_2 + \dots + s_N \\ & \text{s.t. } \|A_i\theta + b_i\| \leq \gamma(c_i^\top \theta + d_i) + s_i \\ & \quad s_i \geq 0, \quad \forall i \in I. \end{aligned} \tag{10}$$

When the sum of the infeasibilities is found to be zero, the original problem is feasible, i.e., a solution θ can be found inside the intersection of the second order cones. If any of s_i s is not zero, then the original problem is infeasible. It may be noted that the SOI method is L_1 minimization of the infeasibility variables. Below, we give a very brief introduction to the L_1 norm minimization. Details can be found in [14,15].

2.3 L_1 Minimization

For the residual vector $\mathbf{r} = \mathbf{f}(\boldsymbol{\theta})$ and the parameter vector $\boldsymbol{\theta}$, the L_1 norm minimization problem is given by:

$$\min_{\boldsymbol{\theta}} |r_1| + \dots + |r_N|, \quad (11)$$

where $\mathbf{r} = [r_1, \dots, r_N]^T$. This is called the sum of (absolute) residuals approximation problem. The L_1 norm minimization is robust estimation because it is most robust to outliers among the minimization methods based on convex penalty functions. Its penalty function places relatively large weight on small residuals compared to others such as the quadratic function for least-squares optimization. Therefore, it tends to produce optimal residuals many of which are very small, or even exactly zero. In statistical terms, L_1 optimization can be interpreted as the maximum likelihood estimation with a Laplacian noise density function which has larger tails than the Gaussian function.

2.4 Outlier Selection with Feasibility Test

Both of the feasibility test algorithms shown above provide an estimate $\boldsymbol{\theta}^*$ whether or not the constraints are found to be feasible. If the constraints are feasible, we can employ the outlier removal scheme shown in [14]. In other words, if the maximum residual e_∞ is larger than a threshold predefined in some way, the support set defined in [14] is discarded and the feasibility test is repeated. Now, here is our question. What if we do not adopt the iterative *optimize-and-discard* strategy and simply run a feasibility test once with a constant (e.g., $\gamma := 3$ pixels) as a threshold for maximum residual?

If we are able to get an estimate $\boldsymbol{\theta}^*$ and identify all the outliers by examining their residuals, we may simply run a feasibility test just once for removing outliers and greatly reduce computation time. The minimization of maximum infeasibility is a useful method that tests the feasibility using s and finds the solution $\boldsymbol{\theta}^*$ inside the intersection of convex constraints. However, it does not provide any clue about the feasibility of each of the constraints when the problem is infeasible due to outliers.

The SOI method, on the other hand, allows us to check the feasibility of individual measurement since its associated variable s_i indicates the infeasibility. If the constraints are all feasible, the objective $\sum_i s_i$ is zero and there are no outliers. If not, the measurements with positive infeasibilities for a given threshold γ are considered as outliers and removed. As noted earlier, the SOI method is L_1 minimization of the infeasibility variables. Indeed, it has a desirable property that we are looking for. Given a maximum threshold γ , the SOI method collects as many measurements as possible that satisfy the given constraints to make the objective function as small as possible.

In the following sections, we examine the performance of the SOI method as a global optimizer having the capability of outlier removal.

3 Camera Resectioning

We use synthetic data for examining the performance of the feasibility test algorithms: the minimization of maximum infeasibility and the SOI method. We also compare those

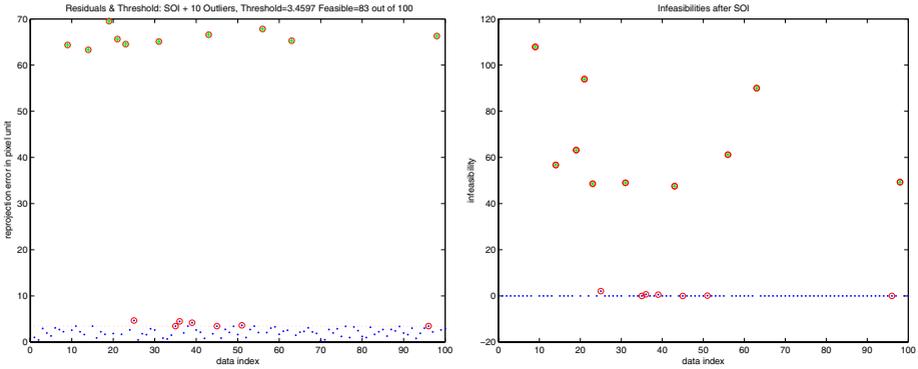


Fig. 1. Left: Residuals of the SOI method. Ten percent of the data set is outliers. Seventeen out of a hundred are classified to be outliers which are denoted by red circles. **Right:** Plot of the infeasibilities. True outliers are found to have very large infeasibilities. Green cross inside the red circle denotes a true outlier.

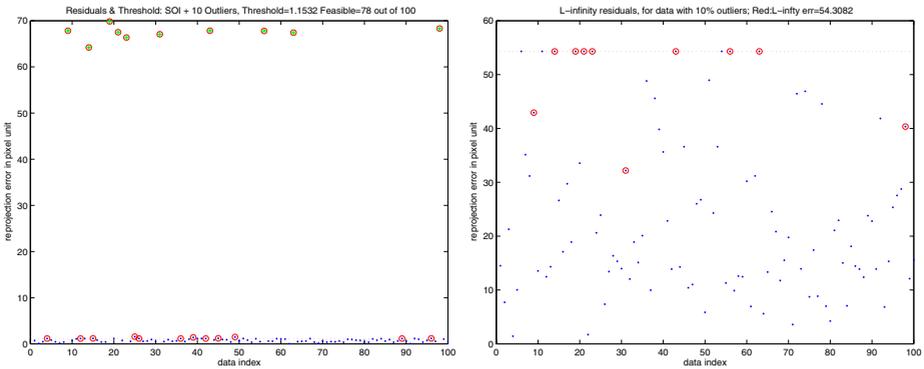


Fig. 2. Left: Residuals of the SOI method (10% outliers). Twenty two, denoted by red circles, out of a hundred are greater than the threshold. The separation of outliers and inliers are much more distinct compared to the result of the L_∞ optimization shown below. **Right:** Residual plot of the L_∞ optimization. The support set is composed of true outliers and inliers.

with the iterative outlier-removal algorithm by Sim and Hartley [14]. For the camera resectioning problem, 100 synthetic data are generated and corrupted with zero-mean Gaussian noise with standard deviation $\sigma = 0.5$ pixel. For outlier generation, we randomly select some measurements and corrupt it with uniform noise with the range of $\pm[5, 50]$ pixels. For testing our algorithms with this type of synthetic data, we may determine the maximum threshold (γ) based on the Gaussian noise distribution. It is our rule of thumb to have γ three times the standard deviation of residuals. The threshold can also be set arbitrarily, e.g., 3 pixels as was done in [11]. For the experiments shown in this section, the L_∞ optimization is carried out on the outlier-free data to find the maximum reprojection error e_∞ , and a multiple of e_∞ is used for γ .

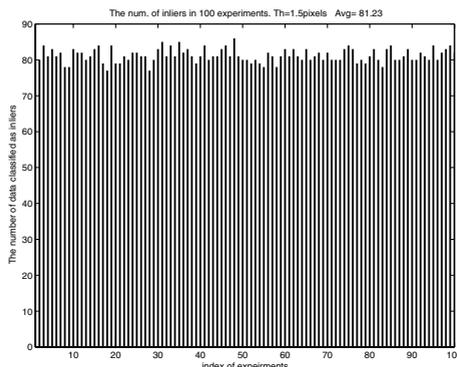


Fig. 3. The number of data points classified as inliers through 100 repeated computations of the SOI method

Figure 1 shows an experiment with the SOI method. The population of outliers is 10% of the data. The left plot shows the residuals and the right does the values of the infeasibility variables s_i s. The residuals above the threshold and the non-zero s_i s are enclosed with red circles, and the true outliers are marked with green cross. Most of the inliers have small residuals and very small values of s_i s ($\approx 10^{-12}$) while the data with large residuals or with substantially positive s_i s are true outliers. We note that the detected outliers always include the true outliers although the SOI method itself provides no explicit guarantee for such performance.

Figure 2 shows two plots, one from the SOI method and the other from the first iteration of the L_∞ outlier removal method by Sim and Hartley [14]. The data set has 10% outliers (ten out of hundred). It can be easily seen that the separation of outliers and inliers are much more distinct with the SOI method. The SOI method finds twenty two outliers including the ten true outliers. The support set of the L_∞ method has ten elements and seven of them are true outliers. Therefore, iterations are required to find all the outliers. As mentioned earlier, the maximum threshold can be determined arbitrarily and it does not have to rely on e_∞ . In this experiment, we set γ to the same value as e_∞ for the purpose of comparison with the L_∞ method.

To show the statistical performance, we run the SOI algorithm a hundred times for 100 randomly generated measurements with Gaussian noise ($\sigma = 0.5$) and 10% outliers. The maximum threshold is fixed to $\gamma := 1.5$ pixels. Figure 3 shows the number of data points identified as inliers. On average, 81 inliers are collected. Since the number of true inliers are 90 of 100, approximately $100 \times 81/90 \approx 90\%$ of true inliers are retrieved. In all the 100 experiments, the true outliers are correctly classified. While the set of detected outliers contains some true inliers, the set of identified inliers never includes any true outliers in any experiment.

Finally we compare the computation times for the SOI method and the iterative L_∞ outlier removal method by Sim and Hartley [14]. To examine the number of repetitions of the feasibility test, the iterative L_∞ outlier removal method is applied to 100 synthetically generated data sets with Gaussian noise ($\sigma = 0.5$) and two outliers. Figure 4 shows the result. For the data set we used, the minimum number of repetitions is 22. The

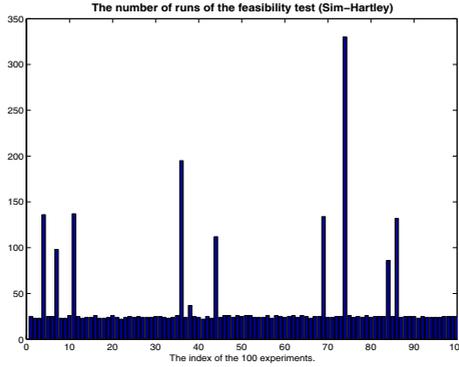


Fig. 4. Plotted is the number of runs of the feasibility test for 100 experiments with the method of Sim and Hartley [14]. The number of runs can be very large sometimes and cannot be expected.

bisection algorithm is run twice to have the maximum residual smaller than the threshold of 2 pixels, and the two outliers are eventually eliminated. The computation time for this case of minimum repetition is 2.6 sec and each feasibility test takes $2.6/22 \approx 0.12$ sec. In contrast, the SOI method needs only one feasibility test. It takes 0.25 sec on average and thus it is ten times ($10.1 \approx 2.6/0.25$) faster than the iterative L_∞ method. In the worst case, the iterative method has 330 repetitions of the feasibility test from 23 runs of the bisection algorithm. It takes 37.8 sec and the computation time is 133 times longer than that of the SOI method.

4 Motion and Structure with Known Rotation

4.1 Synthetic Data

Synthetic data are generated for three views and each view has one hundred points. The data are contaminated with zero-mean Gaussian noise with standard deviation of 0.5 pixels. The maximum residual e_∞ s are found to be in the range of $1.2 \sim 1.8$ pixels after the L_∞ optimization is performed using Kahl's implementation [7]. For generating an outlier, a measurement is randomly selected and the error of $1000e_\infty$ is added to both of the image coordinates. The histogram in Figure 5(Left) shows a result of the SOI method with $\gamma := e_\infty$. To enhance the visibility of the histogram, the residuals e_i s are log-normalized by γ and the histogram is plotted on $\log(e_i/\gamma)$. The origin (the zero point) in the x -axis is the threshold, and all the data located on the left side of the origin are true inliers.

For another experiment, a set of 1000 random data (100 measurements in 10 views) is generated and contaminated with the same Gaussian noise. Then 1% of the data (10 randomly selected data among 1000) are contaminated with random errors of uniform distribution in the range of $[6, 100]$ pixels. The maximum threshold for outliers is set to 2 pixels, i.e., $\gamma := 2$. Figure 5(Right) is the log-residual plot of the result of the SOI method. The detected outliers are 44 and they include all the true outliers. Most of the residuals are on the left side of the origin.

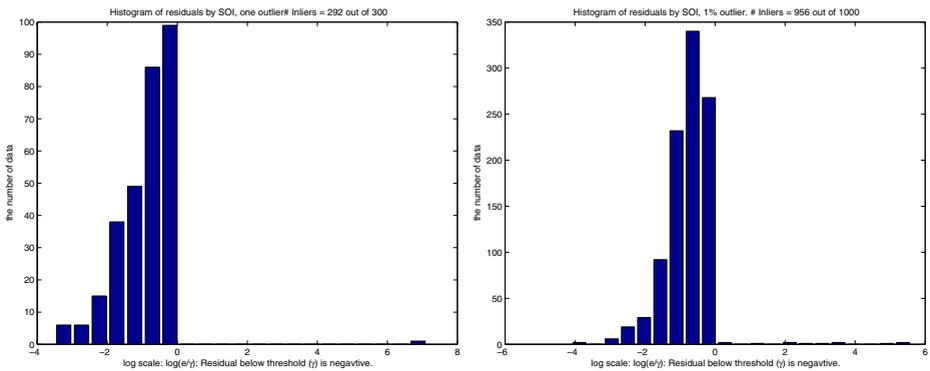


Fig. 5. Left: Histogram of the residuals of the SOI method applied to the data set of 299 inliers and one outlier (100 points in three views). The residuals are log-normalized by γ and the histogram is plotted on $\log(e_i/\gamma)$. **Right:** Histogram of the residuals when the percentage of outliers are 10% (10 outliers among 1000 measurements in ten views).

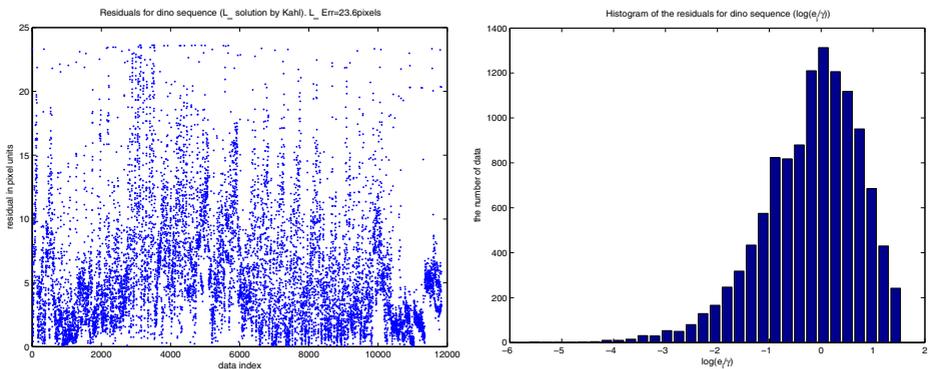


Fig. 6. Left: Residuals obtained by the L_∞ optimization with the help of Kahl’s program [2]. **Right:** Histogram of the residuals $\log(e_i/\gamma)$.

4.2 Real Data

For experiment with real data, we use the dinosaur sequence¹. The data set provides point correspondences and camera matrices. The L_∞ optimization and the SOI method are used to estimate the 3D structure and the translations of 36 cameras under the assumption of known rotation. The maximum threshold of the SOI method for inlier residuals is taken to be $\gamma := 5$ pixels.

Figure 6 shows the residuals and their histogram after the L_∞ optimization is carried out. The L_∞ error, i.e., the maximum error is 23.6 pixels. The histogram of the log-residuals, $\log(e_i/\gamma)$, is almost symmetrically distributed with respect to the threshold.

¹ www.robots.ox.ac.uk/~vgg/

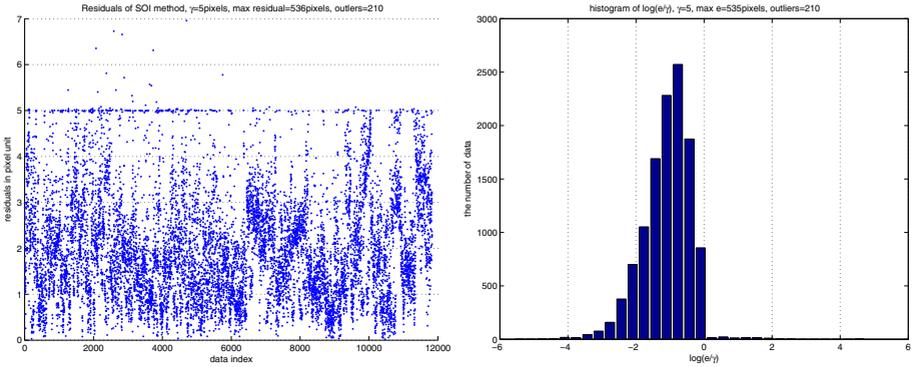


Fig. 7. Left: Residuals obtained by the SOI method ($\gamma := 5$ pixels). **Right:** Histogram of the log-residuals $\log(e_i/\gamma)$.

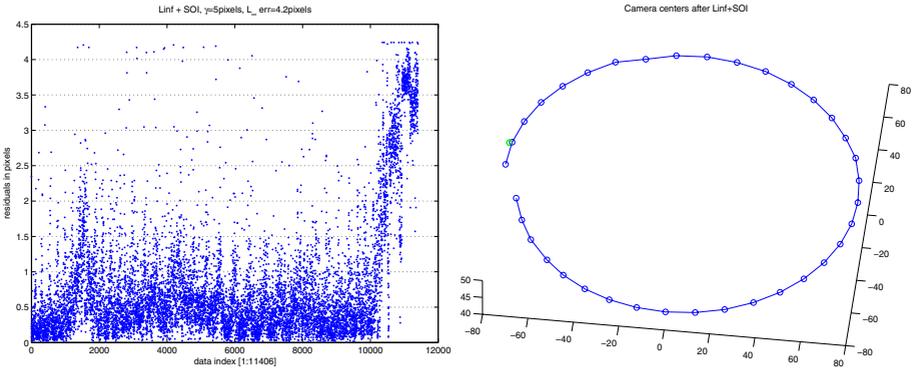


Fig. 8. Left: Residuals of the L_∞ optimization which is preceded by the SOI method. Measurements having residuals larger than $\gamma := 5$ pixels are all removed by the SOI method. **Right:** Plot of the camera centers obtained from the L_∞ optimization.

If outliers are mainly responsible for the error, the iterative outlier removal method shown [14] can be used to improve the result.

Figure 7 shows the plot of residuals and the histogram of log-residuals after the SOI method is performed. The maximum residual is as large as 536 pixels, which is not shown in the plot that focuses on the low range of residuals. Notice that most of the residuals are located below the threshold. We remove 210 measurements among 11832 that have larger residuals than the maximum threshold $\gamma := 5$, and their corresponding points in other views. The number of measurements are reduced to 11406, 96.4% of the total measurements. Figure 11 shows the result after the L_∞ optimization is carried out on the inliers. The L_∞ error is decreased to 4.2 pixels. The lower plot shows the 3D locations of the 36 camera centers. We also run the method of Sim and Hartley [14] until the maximum residual become less than the threshold $\gamma := 5$. Our implementation

requires 16 iterations of the bisection algorithm to remove the outliers and one more bisection algorithm for the final optimization. It took 66283 seconds in total. Our SOI method and the final optimization took 6131 seconds. Thus, our approach is approximately ten times faster than the iterative L_∞ method for the dino data set. Of course, there will be some ways that help the iterative method do better. The computation result will also be different from one data set to another. However, one cannot expect how long it will take with the iterative L_∞ method. Contrarily, one execution of the SOI method guarantees that the residuals become less than the threshold.

5 Conclusion

This paper presented the method of minimizing the sum of infeasibilities (SOI) as a non-iterative outlier removal algorithm. We showed that the SOI method had a very interesting property of collecting inliers as many as possible whereas admitting outliers of large residuals. Therefore, we were able to preset the maximum admissible threshold for the feasibility test to sort out outliers. Experiments with synthetic and real data sets showed the performance of the SOI method. In particular, the experiment with the dinosaur sequence revealed that the algorithm was practically useful in discarding potential outliers and collecting inliers. From the result of this research, we propose that the SOI optimization should precede the L_∞ optimization in order to remove any potential outliers in the data set.

References

1. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2004)
2. Kahl, F.: Multiple view geometry and the L_∞ -norm. In: Proc. Int. Conf. on Computer Vision, Beijing, China, pp. 1002–1009 (2005)
3. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: Proc. Int. Conf. on Computer Vision, Beijing, China (2005)
4. Sim, K., Hartley, R.: Recovering camera motion using L_∞ minimization. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006)
5. Aström, K., Enquist, O., Olsson, C., Kahl, F., Hartley, R.: An L-infinity approach to structure and motion problems for 1d-vision. In: IEEE International Conference on Computer Vision (2007)
6. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. In: IEEE International Conference on Computer Vision (2007)
7. Sim, K., Hartley, R.: Removing outliers using the L_∞ norm. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2006)
8. Li, H.: A practical algorithm for L_∞ triangulation with outliers. In: IEEE International Conference on Computer Vision and Pattern Recognition (2007)
9. Olsson, C., Eriksson, A.P., Kahl, F.: Efficient optimization for L_∞ problems using pseudo-convexity. In: IEEE International Conference on Computer Vision (2007)
10. Seo, Y., Hartley, R.I.: A fast method to minimize L_∞ error norm for geometric vision problems. In: IEEE International Conference on Computer Vision (2007)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM, 381–395 (1981)

12. Sturm, J.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software* 11-12, 625–653 (1999)
13. Hartley, R., Kahl, F.: Global optimization through searching rotation space and optimal estimation of the essential matrix. In: *IEEE International Conference on Computer Vision (2007)*
14. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
15. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge (1992)