Convergence of Iteratively Re-weighted Least Squares to Robust M-estimators

Khurrum Aftab Richard Hartley
Australian National University, ACT 2601, Australia
National ICT Australia (NICTA)
khurrum.aftab@anu.edu.au

Abstract

This paper presents a way of using the Iteratively Reweighted Least Squares (IRLS) method to minimize several robust cost functions such as the Huber function, the Cauchy function and others. It is known that IRLS (otherwise known as Weiszfeld) techniques are generally more robust to outliers than the corresponding least squares methods, but the full range of robust M-estimators that are amenable to IRLS has not been investigated. In this paper we address this question and show that IRLS methods can be used to minimize most common robust M-estimators. An exact condition is given and proved for decrease of the cost, from which convergence follows. In addition to the advantage of increased robustness, the proposed algorithm is far simpler than the standard L_1 Weiszfeld algorithm. We show the applicability of the proposed algorithm to the rotation averaging, triangulation and point cloud alignment problems.

1. Introduction

Robustness to outliers is one of the desirable feature while selecting or proposing an algorithm. Generally, robustness is achieved by either introducing a pre-processing step of outliers removal or by proposing a complicated optimization strategy. However, in this paper we minimize the trade-off of simplicity and robustness by proposing a very simple IRLS algorithm to minimize robust cost functions such as the Huber function, the Pseudo-Huber function [17], etc. The proposed algorithm can also be used to minimize several other cost functions that satisfy certain properties that are identified at a later stage in the paper. There are many applications of the proposed algorithm but here we only consider the problems of rotation averaging [2, 13, 12], triangulation [3, 16, 30] and point cloud alignment [18]. These problems are solved by iteratively solving their weighted least squares forms.

In the context of distance minimization problem, where we seek a point that is closest to a given set of points $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$, we are interested in the following cost function,

$$\min_{\mathbf{x}} \sum_i h(d(\mathbf{x}, \mathbf{y}_i)) \; , \tag{1}$$

where $h(\cdot)$ is a desired robust function and $d(\cdot,\cdot)$ is a distance function.

Generally, a least squares criterion is minimized where $h(\cdot)$ is a square function. Compared to the minimization of other cost functions, the minimization of a least squares cost function has the advantage of being very simple. However, it suffers from a drawback when it comes to robustness against outliers; even a single outlier of high magnitude can deviate the solution from the ground truth value. Thus, a least squares technique is normally preceded by an outlier removal step to give a robust solution.

Due to this, since the last decade, researchers have been exploring techniques for the minimization of other robust cost functions such as the L_1 cost function, the Huber function, the Pseudo-Huber function, etc. A large number of techniques that exist in the literature use complex optimization strategies to minimize these robust cost functions. On the contrary, we propose a very simple optimization strategy that iteratively solves a weighted least squares cost function to find a robust solution of a problem. Thus, in addition to the advantage of being robust, the proposed technique inherits all of the advantage of the corresponding least squares approach.

In this paper, we show that the minimization of (1) can be achieved by iteratively minimizing the following cost function,

$$\min_{\mathbf{x}} \sum_i w_i \, d(\mathbf{x}, \mathbf{y}_i)^2 \; .$$

where w_i is a scalar weight value associated with its corresponding distance term $d(\mathbf{x}, \mathbf{y}_i)$. We show that for certain value of weights, the above algorithm converges to the minimum of a desired cost function.

The algorithm presented in this paper is inspired by the Weiszfeld algorithm [31] and its generalizations [2, 7, 6, 5], specifically the L_q Weiszfeld algorithm that solves the problem of finding the L_q mean of a set of point in $\mathbb{R}^N$, more generally on a Riemannian manifold. Given a set of points, the L_q mean minimizes the sum of the q-th power of distances to the given points. It is proved in [2] that the IRLS algorithm, for certain value of weights, converge to the L_q solution. The L_q Weiszfeld algorithm can be thought as a special case of the proposed IRLS algorithm, where $h(\cdot)$ in (1) is simply an identity map.

Compared to the L_q Weiszfeld algorithm, the proposed algorithm not only gives the advantage of minimizing a variety of robust cost functions but also simplifies the algorithm to a great extent. In the cases of the L_q Weiszfeld algorithms [2, 3] the update functions are not defined everywhere, specifically at points where the distance function is zero. These conditions have been dealt with explicitly in [2, 3] by redefining the update function for these points. However, in the case of the proposed IRLS algorithm the update function, for most of the robust functions, is defined everywhere and these conditions do not occur. As a result, the proposed algorithm is far simpler than the L_q or L_1 Weiszfeld algorithm. It is shown at the end of section 3 how this problem is avoided by using other robust M-estimators.

In addition to the theoretical proof of convergence of the IRLS algorithm to a desired minimum, we show the applicability of the proposed algorithm in computer vision by solving three problems: Firstly, we solve the single rotation averaging problem [2, 13, 12, 32, 1], where a set of relative rotations are averaged to give a better estimate of the actual rotation; Secondly, we solve the triangulation problem [3, 16, 30] where, given a set of projections of a 3D point, we seek a closest-point to the back-tracked lines that best represents the 3D point; Lastly, we solve the problem of point cloud alignment [18], where it is desired to find a transformation that best aligns two sets of 3D points. We specifically compare the results of the Pseudo-Huber, L_1 and L_2 methods. In all the cases, our experimental results confirm the fact that the proposed Pseudo-Huber methods give superior results to the L_2 algorithms and are as robust to outliers as the L_1 algorithms.

In summary, the proposed algorithm uses an IRLS approach to minimize a desired robust cost function. Thus, the proposed algorithm is simple to understand and trivially fits into the existing literature of least squares techniques. An existing least squares implementation of a problem can easily be modified to minimize a desired cost function.

2. The IRLS algorithm

In most general terms, the IRLS algorithm seeks to minimize a cost function

$$C_h(\mathbf{x}) = \sum_{i=1}^{k} h \circ f_i(\mathbf{x}) \tag{2}$$

where each f_i is a function defined on some domain, taking non-negative real-values, and h is some function chosen to make the result robust to outliers in the values $f_i(\mathbf{x})$. As a concrete example, the reader may think of $f_i(\mathbf{x})$ as being the distance of $\mathbf{x}$ from a point $\mathbf{y}_i \in \mathbb{R}^N$ and h as being a Huber cost function. In this case the problem seeks the point $\mathbf{x}^*$ that minimizes the Huber distance of $\mathbf{x}^*$ to the points $\mathbf{y}_i$. This is a generalization of the Fermat-Weber problem [7, 2].

Starting from some initial value $\mathbf{x}^0$, the algorithm solves a sequence of weighted least squares problems

$$x^{t+1} = \operatorname*{argmin}_{\mathbf{y}} \sum_{i=1}^{k} w_i(\mathbf{x}^t)\, f_i(\mathbf{y})^2 \tag{3}$$

where $w_i(\mathbf{x}^t)$ is some (real-valued) "weighting function", used to recompute weights at each step. For brevity, $w_i(\mathbf{x}^t)$ will be denoted by w_i^t. Note specifically that the weights w_i^t are computed from the previous value $\mathbf{x}^t$, and are then not involved in the minimization over $\mathbf{y}$, which is then simply a weighted least-squares problem. The IRLS algorithm is most suited, therefore, to situations where the weighted least-squares problem is easily solvable, perhaps in closed form.

A basic requirement is that the weights are chosen such that indeed the iterative procedure converges to a minimum of (2). This suggests the condition that $\nabla_{\mathbf{x}} \left(h \circ f_i(\mathbf{x})\right) = 0$ if and only if $\nabla_{\mathbf{x}} \left(w_i(\mathbf{x}^t) f_i(\mathbf{x})^2\right) = 0$, which leads to a formula for the weights:

$$w_i(\mathbf{x}) = \frac{h'(f_i(\mathbf{x}))}{2 f_i(\mathbf{x})} \;. \tag{4}$$

2.1. Convergence

Of course, simply choosing the weights as in (4) does not ensure convergence of the iterations (3) to a minimum of the cost function (2). The most important requirement is that each iteration actually reduces the cost (2). Since each iteration (3) minimizes quite a different cost from (2), it is not clear that this will happen. This question will now be considered.

To state the most general condition, we require the concept of supergradients. A supergradient of a function $h : \mathbb{R} \rightarrow \mathbb{R}$ at a point c is a value h^s such that $h(d) \le h(c) + (d-c)\, h^s(c)$ for any point d. A concave function h has a supergradient at every interior point, and if the function is differentiable, then the supergradient at a point is unique and equal to the derivative.

Lemma 2.1. *Let $g(x)$ be a concave function defined on a subset D of real numbers and $g^s(c_i)$ be a supergradient at c_i. Let c_i and d_i in D satisfy*

$$\sum_{i=1}^{k} d_i\, g^s(c_i) \le \sum_{i=1}^{k} c_i\, g^s(c_i) \;.$$

Then
$$\sum_{i=1}^{k} g(d_i) \leq \sum_{i=1}^{k} g(c_i) \ .$$
If the first inequality is strict, so is the second.

Proof. Since g^s is a supergradient, $g(d_i) \leq g(c_i) + (d_i - c_i)\, g^s(c_i)$ for all i. Summing over i gives
$$\sum_{i=1}^{k} g(d_i) \leq \sum_{i=1}^{k} g(c_i) + \sum_{i=1}^{k} (d_i - c_i) g^s(c_i) \ .$$
The last sum is non-positive or negative by hypothesis, completing the proof. □

Applying this lemma to the case where $c_i = f_i(\mathbf{x}^t)^2$, $d_i = f_i(\mathbf{x}^{t+1})^2$ and $g(x) = h(\sqrt{x})$ leads immediately to the following condition.

Lemma 2.2. *Let $h(\sqrt{x})$ be a concave function for $x \geq 0$ and $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$ be two values such that*
$$\sum_{i=1}^{k} w_i^t \, f(\mathbf{x}^{t+1})^2 \leq \sum_{i=1}^{k} w_i^t \, f(\mathbf{x}^t)^2$$
where w_i^t is given by (4). Then
$$\sum_{i=1}^{k} h \circ f_i(\mathbf{x}^{t+1}) \leq \sum_{i=1}^{k} h \circ f_i(\mathbf{x}^t) \ .$$
If the first inequality is strict, so is the second.

This lemma gives the condition ($h(\sqrt{x})$ must be concave) under which a decrease in the weighted L_2 cost function in (3) implies a decrease in the robust cost function (2). This result was proved for the case of L_1 minimization ($h(x) = x$) in [31] and for L_q minimization in [2]. However, to our knowledge, this condition not been noted previously for the general case.

Descent of the cost function is not sufficient to ensure convergence of the algorithm. Conditions to ensure convergence were, however, given in [2] (for the L_q case), essentially equivalent to the Global Convergence Theorem ([22]). In particular, the IRLS algorithm will converge to the set of minima of (2) provided the following conditions are satisfied.

1. Functions f_i and h are continuous (and hence so is C_h).
2. $h(\sqrt{x})$ is concave and differentiable for $x \geq 0$.
3. The function
$$W(\mathbf{w}) = \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{k} w_i \, f_i(\mathbf{x})^2$$
is continuous as a function of the weights $\mathbf{w} = (w_1, \ldots, w_k)$.
4. The sublevel set $\{\mathbf{x} \mid C_h(\mathbf{x}) \leq C_h(\mathbf{x}^0)\}$ is bounded (and hence compact).

It is also implicitly assumed that the weighted least squares problem (3) is solvable. Under these conditions, the sequence of iterates will converge to the set of critical points of (2). If there are a countable number of such critical points, then the iterates will converge to a single critical point. In particular, if (2) is convex, then the IRLS algorithm will converge under these conditions to the global minimum. Given the result of lemma 2.2, proof of convergence follows closely along the lines of the convergence proof in [2].

3. Robust Cost Functions

It was seen in the previous section that the critical condition for the IRLS algorithm to work with robust M-estimators (the function h) is that $h(\sqrt{x})$ should be differentiable at all points and concave. It will be seen next that this condition holds for a large number of robust estimators. In particular, it is true of all the robust cost functions given in [17], section A6.8. These cost functions are shown in fig. 1. Also shown are plots of the functions $h(\sqrt{x})$, from which they are easily seen to be concave.

Certain properties of the various robust cost functions will now be given.

1. L_1 cost function. The functions $h(x)$ is not differentiable at $x = 0$, so the cost function is not differentiable everywhere. Furthermore $h(\sqrt{x})$ is not differentiable at $x = 0$, so the weight $w(x)$ is not defined there. This causes significant theoretical and practical difficulties if for some intermediate t, $f_i(\mathbf{x}^t) = 0$ for some i. For this reason, the algorithm cannot be guaranteed to converge to the minimum cost solution, as noted by Weiszfeld [31], but may get stuck at a point where $f_i(\mathbf{x}) = 0$.

2. L_q cost function. Although the cost function is differentiable when $x = 0$, it is not differentiable (the gradient is infinite). Consequently the weight is infinite at such a point. Once again, this causes significant difficulties, as pointed out most particularly in [3]. That paper showed that in the problem of finding the L_q closest point to a set of affine subspaces in $\mathbb{R}^N$, there is at least one non-optimal point on each subspace or each intersection of subspaces to which the IRLS algorithm may converge.

3. Huber and Pseudo-Huber: The function $h(x)$ is differentiable for $x \geq 0$. and the weights are well-defined everywhere. This is the ideal situation, and convergence to a minimum is assured. If the problem is convex, the global minimum will be found.

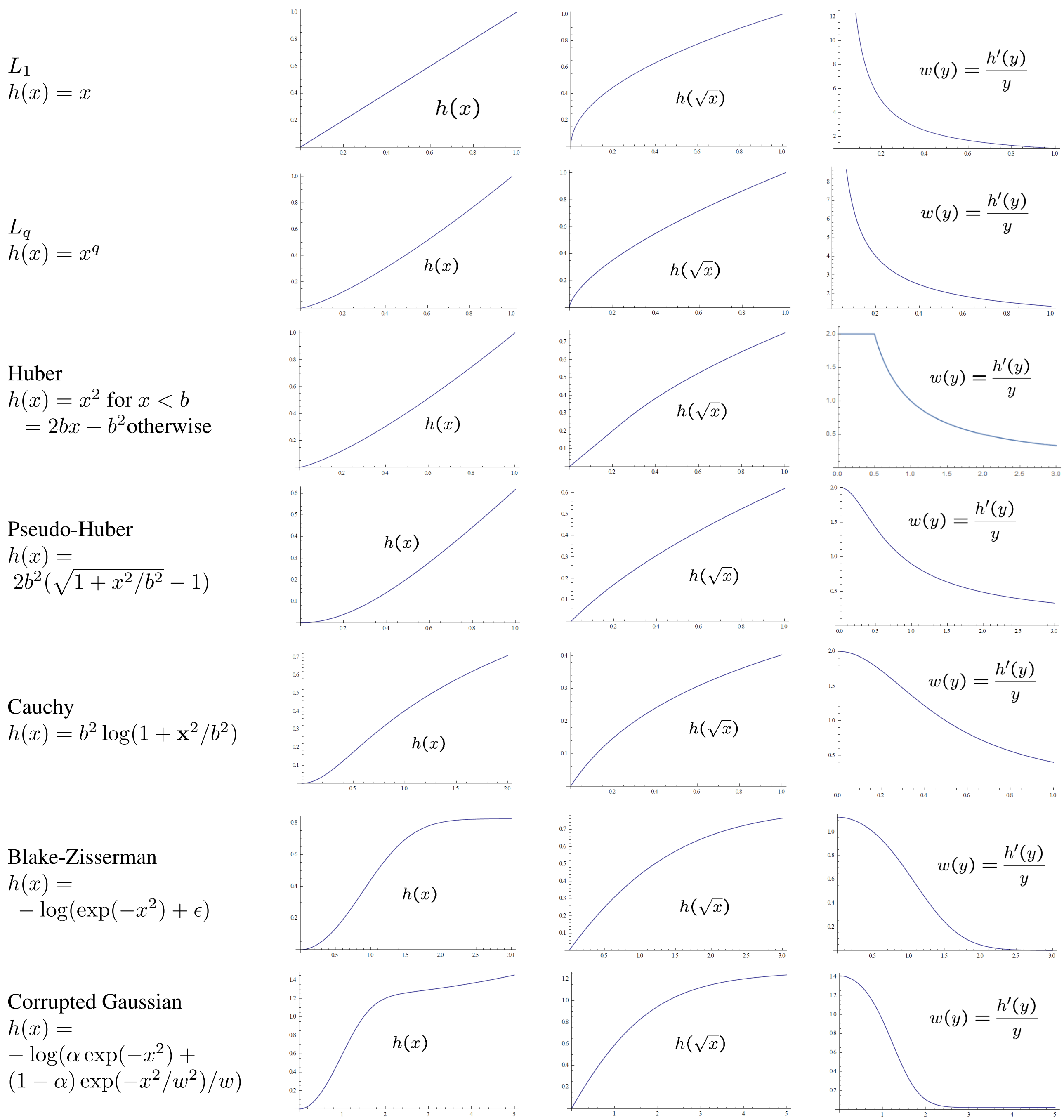


Figure 1. *Various robust cost functions. Observe that in all cases, $h(\sqrt{x})$ is concave. This implies that the IRLS step will always reduce the cost, according to lemma 2.2.*

4. Cauchy, Blake-Zisserman, Corrupted Gaussian: In all of these cases the function $h(x)$ is differentiable everywhere for $x \geq 0$. The algorithm will therefore converge to a critical point. As noted, however, in [17], the non-convexity of the function $h(x)$ can lead to local minima in the total cost function (2), so it is not certain that the global minimum will be found.

As this shows, a weakness of the IRLS algorithm for some robust cost functions, such as L_1 and L_q is that the weights may become infinite (undefined) at some points. Although it is possible to define the update step in a continuous way, the algorithm will still be stuck if an iterate hits one of these points. Analysis of this effect, and convergence of the Weiszfeld (IRLS) algorithm has led to many papers, such as [7, 6, 5]. Furthermore, attempts to analyze this difficulty have a long history of proofs and counterexamples to incorrect claims; see the discussion in [7], and counterexamples in [9, 8]. However, the difficulties, both practical and theoretical disappear if (for instance) a Huber or Pseudo-Huber cost function is used.

Another common approach is to cap the weights at some large finite value, but this is generally proposed as an ad-hoc procedure, with no analysis of what cost is then actually being minimized. The use of such cost functions as Huber or Pseudo-Huber shown above avoids the problems in a principled way, with theoretic justification and a guaranteed decrease in cost and conditions for convergence.

4. Application I: Single Rotation Averaging

In our experiments, we concentrate on IRLS algorithms minimizing the Pseudo-Huber cost function, but the general IRLS method can be applied to any of the other robust cost functions.

In this section we show the applicability of the proposed algorithm to the problem of single rotation averaging [2, 15, 13] where several noisy estimates of a relative rotation are averaged to give a better estimate of the rotation. The problem of rotation averaging has applications to structure and motion [24, 28, 14, 19, 26, 20] and to non-overlapping camera calibration [11]. An IRLS technique to find an L_q mean of a set of rotations has been proposed in [2]. However, we are interested in the minimization of the Pseudo-Huber function by using the IRLS approach.

Given a set of rotations $\{\mathtt{R}_1, \mathtt{R}_2, \ldots, \mathtt{R}_k\}$, we are interested in finding a rotation that minimizes the following function,

$$\min_{\mathtt{S} \in \mathrm{SO}(3)} \sum_{i=1}^{k} h(d(\mathtt{S}, \mathtt{R}_i)) \ ,$$

where $h(\cdot)$ is some robust cost function and $d(\cdot, \cdot)$ is a geodesic distance function. We are interested in the geodesic metric $\theta = d_{\angle}(\mathtt{R}, \mathtt{S})$, which is the angle of the rotation $\mathtt{RS}^{-1}$ [15].

Since the space of rotations can be endowed with a Riemannian metric, we transition back and forth between the rotation manifold, and its tangent space centered at the current estimate via the exponential and logarithm maps. The Riemannian logarithm and exponential maps may be written in terms of the matrix exponential and logarithm as follows.

$$\exp_{\mathtt{S}}([\mathbf{v}]_\times) = \exp([\mathbf{v}]_\times)\mathtt{S}$$
$$\log_{\mathtt{S}}(\mathtt{R}) = \log(\mathtt{RS}^{-1})$$

where $\log$ and $\exp$ (without subscripts) represent matrix exponential and logarithm. The matrix exponential of a skew-symmetric matrix, denoted by $[\cdot]_\times$, may be computed using the Rodrigues formula [17].

In terms of the matrix exponential and logarithms the update of the IRLS algorithm may then be written as

$$\mathtt{S}^{t+1} = \exp\left(\frac{\sum_{i=1}^{k} w_i^t \log(\mathtt{R}_i(\mathtt{S}^t)^{-1})}{\sum_{i=1}^{k} w_i^t}\right)\mathtt{S}^t, \qquad (5)$$

where $\mathtt{S}^t$ is an estimate of the mean at iteration t, and w_i^t is given by (4), depending on the specific M-estimator being used.

For a more complete discussion of metrics on rotation space and rotation averaging, see [15].

Experimental Results: In the rest of the section we show experimental results of the rotation averaging problem on the NotreDame set. The NotreDame set [29] consists of 595 images of $277,887$ points. We only consider those image pairs that have more than 30 points in common. The total number of image pairs that shared more than 30 points are $42,621$. For each image pair, we estimate essential matrices by using a fast five-point algorithm [25]. From each Essential matrix we extracted cheirally correct relative rotation and translation. Only those solutions were retained that fitted well for another 3 points. Several sets of rotations were obtained by using several subsets of 5 points. These rotations were then averaged to find their mean. A closed form L_2 chordal mean [15] was used as a starting point for our algorithms.

In order to show the robustness of the method against outliers we modify some percentage of image point correspondences to represent outliers and estimate a set of rotations from these correspondences using the procedure described before. We then apply the averaging algorithms to perform averaging of these rotations and compute errors with respect to the known ground truth. We compare the results of the proposed Pseudo-Huber algorithm (for different values of threshold) with the L_q rotation averaging, for $1 \le q < 2$ and the L_2 rotation averaging algorithms. It evident from fig. 2 that the results of the Pseudo-Huber averaging, for some value of threshold, are as robust to outliers as the L_1 averaging algorithm. Furthermore, as expected, the L_2 algorithm is the least robust to outliers. Thus, in the presence of outliers the minimization of the Pseudo-Huber cost function is recommended because of its simplicity and robustness to outliers.

5. Application II: Closest-Point to Affine Subspaces

In this section we solve the problem of finding a closest-point to a set of affine subspaces [3] by using an IRLS algorithm and show its applicability to the triangulation problem [16, 30]. Given a set of affine subspaces $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$,

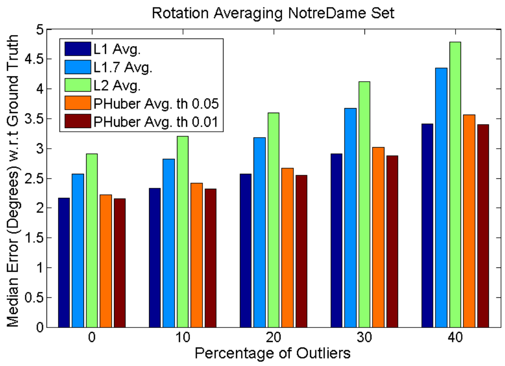


Figure 2. *Single Rotation Averaging: The above figure shows results of the Pseudo-Huber Rotation Averaging (for different values of threshold) and the L_q rotation averaging for $1 \leq q \leq 2$. The rotation averaging is performed on the NotreDame set. We test the algorithms in the presence of different percentage of outliers in the data set. We modifying some percentage of image point correspondences to represent outliers. The results have shown that for certain threshold values the Pseudo-Huber minimization gives as robust results as the L_q averaging for $q = 1$. Furthermore, the L_2 averaging algorithm is the least robust to outliers. Therefore, the Pseudo-Huber algorithm is recommended because of the its simplicity and robustness to outliers.*

in $\mathbb{R}^N$ we seek a point $\mathbf{X}$ such that,

$$\min_{\mathbf{X}\in\mathbb{R}^N} \sum_{i=1}^{k} h(d(\mathbf{X}, \mathcal{S}_i)) ,$$

where $h(\cdot)$ is a desired robust cost function and $d(\mathbf{X}, \mathcal{S}_i)$ is the orthogonal distance of a point $\mathbf{X}$ from $\mathcal{S}_i$. As shown in [3], the corresponding weighted least-squared problem (3) has a closed-form solution, and so solving this problem using robust M-estimators of the type considered in this paper is straightforward.

Triangulation: We use the proposed algorithm to solve the triangulation problem [16, 30]. Given two or more images of a scene, triangulation is a process of determining a point in $3D$ space from its image points. Each image point corresponds to a line in $3D$ space, passing through the center of the camera and intersecting the image plane at the point. Thus, if a camera matrix and an image point is known, a line from the center of camera and passing through the image point can easily be constructed. Ideally, all the back-tracked lines should intersect at a single point but due to noise in image points these lines do not intersect at a single point.

We test the proposed algorithm on a well know dinosaur dataset. This dataset contains a collection of 4983 track points that are tracked over a set total of 36 images. Here we only consider the track points that are visible in more than 10 images. Thus, a minimum of 10 lines are available

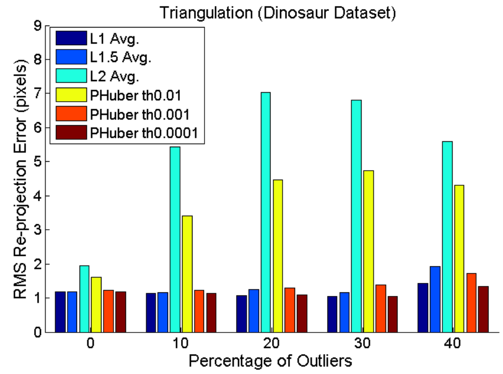


Figure 3. *Triangulation: The above figure shows a plot of re-projection errors in the presence of different percentage of outliers in the dinosaur dataset. The plot shows that for some values of threshold the Pseudo-Huber minimization gives as superior results as the L_q minimization for $q = 1$. Furthermore, as expected the robustness the Pseudo-Huber algorithm to outliers decreases with an increase in the threshold value.* **Note:** *the RMS re-projection error is computed without using the modified image point correspondences, that is the outliers.*

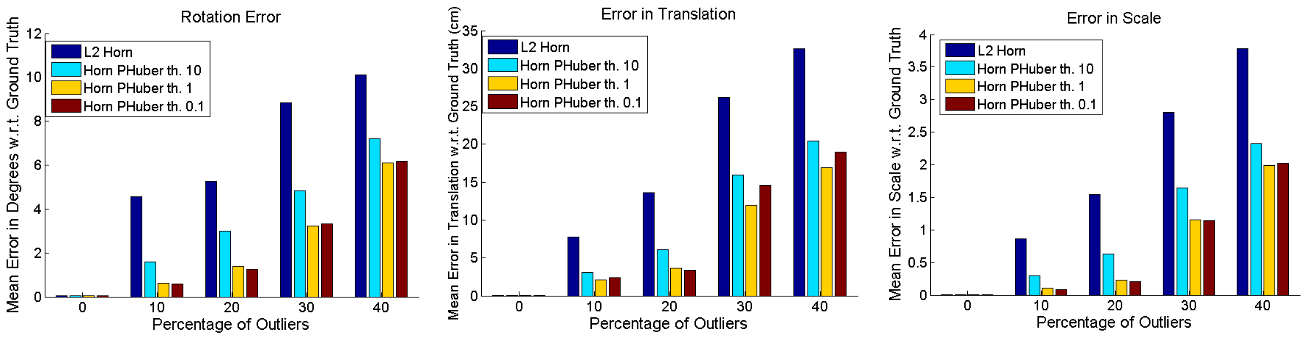
to perform triangulation. We take the L_2-closest-point as a starting point for the algorithms.

The measure of accuracy for reconstructed 3D points is taken to be root mean square (RMS) of the L_1-mean of the re-projection errors, that is the L_1-mean of the distance between reprojected points and measured image points for all the views in which that point is visible. For n reconstructed points $\mathbf{X}_j$, visible in k_j views, the RMS error is computed as follows:

$$e_{\text{rms}} = \sqrt{\textstyle\sum_{j=1}^{n} e_j^2 / n} ,$$

where $e_j = \sum_{i=1}^{k_j} d(\mathbf{x}_{ij}, \mathbf{x}'_{ij})/k_j$, and $\mathbf{x}'_{ij}$ is the measured image point and $\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ is the reprojected point. Note that the error reported here is different from the error being minimized by the proposed algorithm.

Results: In our experiments we modify some percentage of the image points corresponding to a 3D point to represent outliers. We compared the results of the Pseudo-Huber algorithm with the L_q-closest-point method, for $1 \leq q \leq 2$. Furthermore, the RMS re-projection error is computed without using the modified image point correspondences, that is the outliers. Our experimental results in fig. 3 show that the Pseudo-Huber method is as robust to outliers as the L_1 method. In addition to robustness to outliers, the Pseudo-Huber minimization has an advantage of being simpler. Furthermore, our results show that the L_2 technique is very sensitive to outliers. Thus, in the presence of outliers minimization of the Pseudo-Huber cost function is recommended.



(a) Mean Error Rotations (Degrees) (b) Mean Error Translation (cm) (c) Mean Error Scale

Figure 4. *Point Cloud Alignment: The above figures show plots of mean errors in rotation, translation and scale parameters for the the Pseudo-Huber algorithm and the L_2 algorithm in the presence of different percentage of outliers in a synthetic dataset. The left figure is a plot of error in rotation where error is computed as the angle (in degrees) between estimated and the ground truth rotations. The figure in the middle is a plot of error in translation which is measured by computing the magnitude (in cm) of the difference of the estimated and ground truth translation vectors. The right figures shows a plot of error in scale, computed by simply taking the absolute value of the difference in scale values. It is evident from the above plots that in the presence of outliers the parameters recovered by the proposed Pseudo-Huber algorithm are closer to the ground truth than the parameters recovered by the L_2 algorithm. Thus, the above plots confirm the fact that the Pseudo-Huber minimization gives superior results to the L_2 algorithm in terms of robustness to outliers.*

6. Application III: Point Cloud Alignment

In this section we propose an iterative IRLS algorithm to solve the problem of point cloud alignment. Given two sets of points in $\mathbb{R}^3$, our objective is to find a transformation matrix that best aligns the two sets of points. A classic algorithm to solve the problem in closed-form was proposed by Horn [18]. Horn's algorithm has been used widely to solve the registration problems, for example registration of 3D shapes [4], registration of medial images [23], Iterative Closest-Point (ICP) method [27], range image registration [10, 21] and registration of free-form curves and surfaces [33]. Since Horn's algorithm minimizes a least squares cost function, it is sensitive to outliers. Thus, we use the proposed algorithm to find a robust solution of the problem.

Given a set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ and its corresponding scaled, rotated and translated points $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ in $\mathbb{R}^3$, we want to find a transformation that best aligns the two sets of points. Thus, the minimization problem is,

$$\min_{\mathtt{T}} \sum_{i=1}^{k} h(\|\mathtt{T}\mathbf{x}_i - \mathbf{y}_i\|) \ ,$$

where $h(\cdot)$ is a robust cost function and $\mathtt{T}$ is a transformation matrix that maps points in X to points in Y.

The corresponding weighted least-squares problem has a very simple solution that only requires use of appropriate weight values in the equations listed in [18, Appendix A2]. Correspondingly, extension of this method to minimization of robust cost functions is straightforward.

Experiments: We perform our experiments on a synthetic data of 10 point cloud pairs, with 30 points in each cloud. In order to construct the first set we take a point and add a Gaussian noise of zero mean and standard deviation of 10 cm to it to generate the rest of the points in the set. We then construct a transformation matrix, comprising rotation, translation and scale, and apply the transformation to the first set of points to generate the second set of points. After transforming the points we add some more noise to it and modify some of the points to represent outliers. We generate outliers by adding a large amount of noise to the original points, specifically, outliers follow a Gaussian distribution with zero mean and standard deviation of 30 cm. We take the transformation matrix as the known ground truth. We use the L_2 Horn algorithm and the proposed Pseudo-Huber algorithm to estimate the transformation. The L_2 solution was used as a starting point for our algorithm.

For each point cloud, errors in rotation, translation and scale parameters are computed by using the known ground truth values. Error in rotation is computed by finding the angle of the rotation matrix $\mathtt{R}\,(\mathtt{R}')^T$, where $\mathtt{R}$ and $\mathtt{R}'$ are taken to be the ground truth and estimated rotation matrices. Error in translation is computed by finding the magnitude of the difference of the estimated and the ground truth translation vectors. Lastly, error in scale is simply computed by finding the absolute value of the difference of the estimated and ground truth scale values. In the end, we compute the mean of the errors for all point clouds.

Our experimental results in fig. 4 confirm that the proposed Pseudo-Huber technique is more robust to outliers than the L_2 technique, that is the Horn's algorithm. Thus, in the presence of outliers it is recommended to use the proposed Pseudo-Huber algorithm to register points robustly.

7. Conclusion

In conclusion, we identified the convergence conditions and gave a theoretical proof of convergence of IRLS method to a minimum of a desired robust M-estimator such as the Huber function, the Pseudo-Huber function, etc. Since a desired cost function is minimized by alternatively minimizing a weighted least squares cost function, the proposed algorithm is simple to understand and easy to code. Our experimental results for rotation averaging, triangulation and point cloud alignment confirmed the fact that the Pseudo-Huber minimization gives superior results to both the L_1 and L_2 cost functions. Thus, in the presence of outliers the minimization of robust cost functions such as the Pseudo-Huber or Huber function is recommended.

References

[1] B. Afsari. Riemannian L^p center of mass: Existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 2011.

[2] K. Aftab, R. Hartley, and J. Trumpf. Generalized Weiszfeld algorithm for Lq optimization. in preparation.

[3] K. Aftab, R. Hartley, and J. Trumpf. Lq-closest-point to affine subspaces using the Generalized Weiszfeld algorithm. in preparation.

[4] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Proc. Robotics-DL tentative*. International Society for Optics and Photonics, 1992.

[5] J. Brimberg. Further notes on convergence of the Weiszfeld algorithm. *Yugoslav Journal of Operations Research*, 13(2):199–206, 2003.

[6] J. Brimberg and R. Chen. A note on the convergence in the single facility minisum location problem. *Computers Math. Applic.*, 35(9):25–31, 1998.

[7] J. Brimberg and R. F. Love. Global convergence of a generalized iterative procedure for the minisum location problem with lp distances. *Operations Research*, 41(6):1153–1163, 1993.

[8] L. Cánovas, R. Cañavate, and A. Marín. On the convergence of the Weiszfeld algorithm. *Math Prog. Ser. A*, 93:327–330, 2002.

[9] R. Chandrasekaran and A. Tamir. Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. *Math Prog.*, 44:293 – 295, 1989.

[10] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 1992.

[11] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley. Rotation averaging with application to camera-rig calibration. In *Proc. Asian Conference on Computer Vision*, 2009.

[12] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi. The geometric median on riemannian manifolds with applications to robust atlas estimation. *Neuroimage*, 2009.

[13] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the Weiszfeld algorithm. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[14] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[15] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 2013.

[16] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 1997.

[17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision – 2nd Edition*. Cambridge University Press, 2004.

[18] B. K. P. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 1988.

[19] F. Kahl. Multiple view geometry and the L_∞-norm. In *Proc. International Conference on Computer Vision*, 2005.

[20] R. Kaucic, R. I. Hartley, and N. Y. Dano. Plane-based projective reconstruction. In *Proc. International Conference on Computer Vision*, 2001.

[21] J. Luck, C. Little, and W. Hoff. Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In *Proc. IEEE International Conference on Robotics and Automation*, 2000.

[22] D. G. Luenberger. *Linear and nonlinear programming*. Springer, 2003.

[23] J. Maintz and M. A. Viergever. A survey of medical image registration. *Medical image analysis*, 1998.

[24] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[25] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[26] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery. In *Proc. International Conference on Computer Vision*, 2001.

[27] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. IEEE International Conference on 3-D Digital Imaging and Modeling*, 2001.

[28] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc. ACM SIGGRAPH*, 2006.

[30] W. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon. Bundle adjustment for structure from motion. In *Vision Algorithms: Theory and Practice*. Springer-Verlag, 2000.

[31] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 1937.

[32] L. Yang. Riemannian median and its estimation. *LMS Journal of Computation and Mathematics*, 2010.

[33] Z. Zhang. Iterative point matching for registration of freeform curves and surfaces. *International journal of computer vision*, 1994.