

# Fast General Norm Approximation via Iteratively Reweighted Least Squares

Masaki Samejima<sup>(✉)</sup> and Yasuyuki Matsushita

Graduate School of Information Science and Technology,  
Osaka University, Suita, Osaka, Japan  
{samejima,yasumat}@ist.osaka-u.ac.jp

**Abstract.** This paper describes an efficient method for general norm approximation that appears frequently in various computer vision problems. Such a lot of problems are differently formulated, but frequently require to minimize the sum of weighted norms as the general norm approximation. Therefore we extend Iteratively Reweighted Least Squares (IRLS) that is originally for minimizing single norm. The proposed method accelerates solving the least-square problem in IRLS by warm start that finds the next solution by the previous solution over iterations. Through numerical tests and application to the computer vision problems, we demonstrate that the proposed method solves the general norm approximation efficiently with small errors.

## 1 Introduction

In various tasks in digitally archiving cultural heritages including 3D reconstruction [1, 2], numerical optimization plays a central role. More specifically, we often optimize some  $\ell_p$ -norm of the cost vector (equivalently, its  $p$ -th power  $\ell_p^p$ ) or the combination of different vector norms. For example, in compressive sensing, an unconstrained form of Lasso (least absolute shrinkage and selection operator) [3]

$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_1 \quad (1)$$

is used for reconstructing sparse signal  $x$  while ensuring data fitting. As another example, Tikhonov regularization (or ridge regression) [4],

$$\min_x \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2, \quad (2)$$

appears in image restoration, super-resolution, and image deblurring. These objective functions can be further augmented by additional  $\ell_p$ -norm terms that represent further constraints for particular problems. For example, the elastic net [5] is defined as

$$\min_x \|Ax - b\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|x\|_2^2 \quad (3)$$

by regularizing the solution using both  $\ell_1$ - and  $\ell_2$ -norm terms. Some special cases are known to have closed-form solutions, *e.g.*, the minimizer of Tikhonov

regularization (2) is given by  $\hat{x} = (A^T A + \lambda \Gamma^T \Gamma)^{-1} A^T b$ , or can be transformed into a simpler expression, *e.g.*, the elastic net problem (3) can be rewritten as an equivalent Lasso problem (1) with some augmentation [5].

Generally, these  $\ell_p$  ( $p \geq 1$ ) unconstrained minimization problems can be solved by any convex optimization methods<sup>1</sup>. To gain a greater computation performance, typically problem-specific structures are exploited to design a tailored solution method. For example, least-squares problems that consist of  $\ell_2$ -norms can be solved analytically, or for a large-scale problem, conjugate gradient methods [6] are employed for faster computation. When the rank of the design matrix is small enough, randomized singular value decomposition (R-SVD) [7] may be employed for further acceleration. It is also understood that  $\ell_1$  minimization problems can be transformed into a linear programming problem [8], which can be efficiently solved by an interior point method [9]. On the other hand, it is still of broad interest to improve the performance of the general norm approximation problem, because in practical situations there is a strong need for testing with various formulations with different norms in designing computer vision applications. For example, one might initially formulate a regression problem with an  $\ell_2$ -norm but later might add an  $\ell_1$  or  $\ell_2$  regularizer for stabilizing the solution.

This motivates us to develop a fast solver for the generalized norm approximation problem:

$$\min_x \sum_{k=1}^K \lambda_k \|A_k x - b_k\|_{p_k}^{p_k}, \quad (4)$$

where  $k = \{1, \dots, K\}$  is the term index,  $A_k \in \mathbb{R}^{m_k \times n}$  and  $b_k \in \mathbb{R}^{m_k}$  the the design matrix and constant vector that define the  $k$ -th linear objective, and the overall objective function is defined as a linear combination of  $p_k$ -th power of  $\ell_{p_k}$ -norm weighted by  $\lambda_k$ . Our method is built upon a simple yet powerful iteratively reweighted least-squares (IRLS) scheme. In the IRLS scheme, the problem can be reduced to iteratively solving a linear system that is derived as a normal equation of the sum of weighted squares of the terms.

In this paper, we present a fast method for deriving the approximate solution for this problem that outperforms the state-of-the-art solution methods such as [10, 11]. Our method exploits the trait that the solution is gradually updated in an iterative manner in the IRLS scheme, and achieves acceleration by taking the previous estimate as an initial guess at each iteration for an LSQR solver [12]. The proposed method is faster and more stable than the previous state-of-the-art approaches as we are going to see in the experimental validation. In addition, the solution method for the general expression (4) has not been explicitly described in the literature that we are aware of, and we show in this paper that the general form can be solved in a unique manner regardless of the number of terms and diverse  $\ell_p$ -norm objectives and benefit from the proposed method.

---

<sup>1</sup> While it may be still valid even when  $p < 1$ , the problem becomes non-convex when  $p < 1$ ; thus they may be trapped by local minima.

## 2 Related Works

The early studies of IRLS can be found back in 1960's [13], developed for approximating a Chebyshev or  $\ell_\infty$  norm. It has been later extended to approximate a general  $\ell_p$ -norm term [14]. While the early studies focus on convex approximations with  $p \geq 1$  mostly with a single term, later, the focus has been shifted to the case where  $p < 1$  (non-convex cases). The original sparse recovery using the IRLS scheme has been known as FOCUSS [15] prior to that shift, and it has been known useful for robust estimation tasks. With the rise of compressive sensing and sparse recovery, norm approximation with  $p < 1$  has been extensively studied. Chartrand and Yin [16] introduced a regularizer  $\epsilon$ , for augmenting the IRLS weight, that varies from large to small over iterations so that it effectively smooths out the objective function and as a result avoids local minima. Daubechies *et al.* [10] proposed an alternative method for updating the weight over iterations and showed the convergence property in sparse recovery. Candès *et al.* [17] introduced an iteratively reweighted  $\ell_1$  minimization method, which repeatedly solves  $\ell_1$  minimization problem, for further enhancing sparsity. Wipf and Nagarajan [18] provided an extensive analysis on  $\ell_2$  and  $\ell_1$  reweighting schemes [16, 17] and made distinction between separable (*i.e.*, the weighting of a particular coefficient depends only that in previous iteration) and non-separable iterative reweighting schemes. The development of an effective numerical algorithm for sparse recovery is still an active research topic, and there is a broad interest in the area.

With these theoretical and algorithmic development, the IRLS scheme has expanded its application domain. It has been used for various signal processing applications, such as FIR filter design [19], image deblurring with  $\ell_p$ -norm ( $p = 0.8$ ) minimization [20, 21], denoising based on TV regularization [22], and super-resolution [23]. The IRLS scheme has also been used for minimizing nuclear norm [24] and structured sparse recovery [25]. These new applications widen the use of IRLS scheme for even more diverse applications. This paper is motivated by the background that accelerating the general sum of  $\ell_p^p$  terms is urgent because of its increasing need in various computer vision applications.

Because the problem of (4) is unconstrained and convex when  $p_k \geq 1$ , it can be also solved via a family of efficient quasi-Newton methods, such as limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method, although such general convex optimizers are typically not optimal in terms of their performance.

## 3 Fast General Norm Approximation

This section describes the proposed method for general norm approximation using IRLS. In Sect. 3.1, we begin with briefly reviewing the IRLS and show how the generalized form of Eq. (4) can be solved via IRLS. We then describe an acceleration method using LSQR in Sect. 3.2.

### 3.1 IRLS for Norm Approximation Problems

Since 1960's, it has been understood that norm approximation problems can be solved via IRLS [13, 14]. With the IRLS framework, norm approximation problems can be casted to iteratively solving weighted least-squares problem. Let us take an example of minimizing the  $p$ -th power of  $\ell_p$ -norm of a real-valued vector  $(Ax - b)$ :

$$\min_x f(x), \quad f(x) = \|Ax - b\|_p^p = (Ax - b)^T W^T W (Ax - b). \quad (5)$$

The above can be expressed by a weighted squares of the vector as:

$$\min_x f(x), \quad f(x) = \|W(Ax - b)\|_2^2, \quad (6)$$

with a proper diagonal weight matrix  $W$ , whose elements are all non-negative. where  $W$  is a diagonal matrix of weights to be determined in IRLS. The problem of (6) is a quadratic programming; therefore the minimizer  $x^*$  is attained when

$$\frac{\partial}{\partial x} \|W(Ax - b)\|_2^2 = 2p (A^T W^T W Ax - A^T W^T W b) = 0. \quad (7)$$

Therefore, the approximate solution  $x^*$  becomes

$$x^* = (A^T W^T W A)^{-1} A^T W^T W b. \quad (8)$$

In the IRLS scheme, the weight matrix  $W$  is iteratively refined for a more focal estimate. Let  $w_i$  and  $e_i$  denote the  $i$ -th diagonal element of  $W$  and the  $i$ -th element of the residual vector  $e = Ax - b$ , respectively. Since

$$\|Ax - b\|_p^p = \sum_i |e_i|^{p-2} |e_i|^2 = \sum_i w_i^2 |e_i|^2, \quad (9)$$

at each iteration, the weight matrix element  $w_i$  is updated by  $w_i \leftarrow |e_i|^{(p/2-1)}$  if  $e_i \neq 0$  or  $w_i \leftarrow 1/\varepsilon$  otherwise where  $\varepsilon$  is a sufficiently small positive value. Typically, the weight matrix  $W$  is initialized as an identity matrix.

*IRLS for General Norm Approximation.* We have seen how  $\ell_p$ -norm minimization for a single term is achieved by IRLS. We now describe its extension to the multiple terms for applying IRLS to the general norm approximation problem (4). For a general norm approximation problem

$$\min_x f(x), \quad f(x) = \sum_{k=1}^K \lambda_k \|A_k x - b_k\|_{p_k}^{p_k}, \quad (10)$$

there are  $K$  terms that are defined by  $\ell_{p_k}$ -norm, where  $p_k$  may be different across the terms, each weighted by  $\lambda_k$ . With  $K$  weight matrices  $W_k$ , it can be approximated by

$$f(x) = \sum_{k=1}^K \lambda_k (A_k x - b_k)^T W_k^T W_k (A_k x - b_k), \quad (11)$$

in a similar manner to the single term case. From the normal equation of above, the approximate solution can be determined by differentiating  $f(x)$  w.r.t.  $x$  as

$$\frac{\partial f(x)}{\partial x} = \sum_{k=1}^K 2p_k \lambda_k (A_k^T W_k^T W_k A_k x - A_k^T W_k^T W_k b_k) = 0. \quad (12)$$

Therefore, the minimizer  $x^*$  can be obtained by

$$x^* = \left( \sum_{k=1}^K p_k \lambda_k A_k^T W_k^T W_k A_k \right)^{-1} \left( \sum_{k=1}^K p_k \lambda_k A_k^T W_k^T W_k b_k \right). \quad (13)$$

The pseudo-code of IRLS for general norm approximation is summarized in Algorithm 1.

---

### Procedure 1. IRLS for general norm approximation

---

**Input:**  $A_k \in \mathbb{R}^{m_k \times n}$  and  $b_k \in \mathbb{R}^{m_k}$

**Output:** Solution  $x$

```

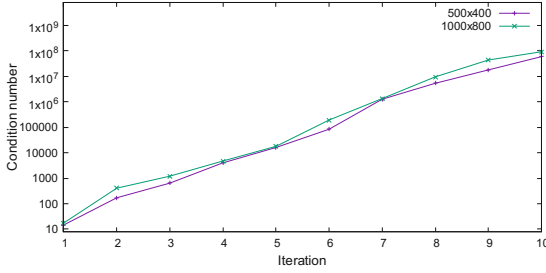
// Initialize the weight matrix
 $W_1, W_2, \dots, W_K \leftarrow I$ 
// Concatenate matrices and vectors;  $A \in \mathbb{R}^{M \times n}$ ,  $b \in \mathbb{R}^M$ ,  $W \in \text{diag}(\mathbb{R}_+^M)$ ,  $M = \sum_k m_k$ 
 $A \leftarrow [A_1^T, A_2^T, \dots, A_K^T]^T$  // Matrix of vertically stacked  $A_1, \dots, A_K$ 
 $b \leftarrow [b_1^T, b_2^T, \dots, b_K^T]^T$  // Vector of vertically stacked  $b_1, \dots, b_K$ 
while  $x$  is not converged do
   $W \leftarrow \text{diag}(\sqrt{\lambda_1 p_1} W_1, \sqrt{\lambda_2 p_2} W_2, \dots, \sqrt{\lambda_K p_K} W_K)$ 
  // Solve the weighted least-squares problem:  $W A x = W b$ 
   $x \leftarrow \text{LeastSquares}(W A, W b)$ 
   $e \leftarrow A x - b$ 
  // Update weight
  for all  $k, n$  do
    if  $e_k(n) \neq 0$  then
       $W_k(n) = 1/|e_k(n)|^{(1-p_k/2)}$ 
    else
       $W_k(n) = 1/\epsilon$  //  $\epsilon$  is a sufficiently small positive value.
    end if
  end for
end while

```

---

## 3.2 Acceleration of IRLS

The major bottleneck of the IRLS algorithm is its need for solving the weighted least-squares problem over iterations until convergence. In particular, when the size of matrix  $A_k$  is large, the computation cost significantly increases if an analytic solution method like Eq. (13) is used. To accelerate the whole algorithm, it is needed to efficiently solve the weighted least-squares problem. We exploit



**Fig. 1.** Growth of the condition number of  $A$  in each iteration. The plots show the average condition numbers over ten trials for each setting.

the fact that the solution  $x$  and weight matrix  $W_k$  are gradually updated over iterations of IRLS and use the previous estimate of  $x$  for efficiently updating the solution  $x$  using LSQR.

Previously, conjugate gradient methods have been applied to the least-squares problem in an iterative framework [25, 26] in a similar spirit. They are effective when the design matrix  $A$  in  $\|Ax - b\|_2^2$  is well-conditioned. Convergence of the conjugate gradient method is analyzed by the relative error [27]:

$$\frac{\|x - x^{(n)}\|_A}{\|x - x^{(0)}\|_A} \leq 2 \left( \frac{\sqrt{\kappa - 1}}{\sqrt{\kappa + 1}} \right)^n, \quad (14)$$

where  $\|\cdot\|_A$  indicates  $A$ -norm,  $\kappa$  is matrix  $A$ 's condition number calculated by  $\kappa = \sigma_{\max}(A)/\sigma_{\min}(A)$ ,  $\sigma_{\max}(A)$  and  $\sigma_{\min}(A)$  are the maximum and the minimum singular values of  $A$ , respectively. When the relative error in the left side of Eq. (14) is large, convergence from  $x^{(0)}$  to  $x^{(n)}$  through  $n$  iterations takes much time. The upper bound of the relative error can be calculated with  $\kappa$ , and a greater  $\kappa$  makes the convergence slower. Unfortunately, for the weighted least-squares problem  $WAx = Wb$  in IRLS, the condition number of matrix  $WA$  naturally increases as the iteration proceeds [26]. To depict this issue, we show a preliminary experiment of running IRLS for solving  $\min_x \|Ax - b\|_2^2$  ten times with randomly generated matrices  $A \in \mathbb{R}^{500 \times 400}$  with vector  $b \in \mathbb{R}^{500}$  and also a larger scale setting,  $A \in \mathbb{R}^{1000 \times 800}$  with vector  $b \in \mathbb{R}^{1000}$ . Figure 1 shows the variation of the average condition number over iterations plotted in a log scale. As seen in the figure, the condition number grows exponentially over iterations, which makes conjugate gradient methods slower and less stable in later iterations.

To overcome this issue, previous approaches use *preconditioning* to yield the equivalent least-squares problem with the small condition number of  $A$  by multiplying a matrix  $P$  called *preconditioner* [25, 26, 28]. However, the preconditioner  $P$  is problem-dependent [25]; therefore, it is not straightforward to incorporate the preconditioned conjugate gradient method into the generalized norm approximation problem.

To avoid these problems, we use the LSQR method [12], which is a stable iterative method for ill-conditioned least-squares problems [29, 30]. In LSQR, the least-square problem is reduced to another least-square problem including a bidiagonal matrix, and the reduced problem is solved by QR factorization. To accelerate the LSQR computation in IRLS framework, we use a “warm start” strategy by taking the previous estimate of the solution as the initial guess for the next iteration. The pseudo-code of LSQR to find a solution  $x^{(i+1)}$  at the  $(i+1)$ -th iteration from  $x^{(i)}$  from the previous iteration in our context is shown in Algorithm 2.

---

**Procedure 2.** LSQR
 

---

**Input:**  $A \in \mathbb{R}^{M \times n}$ ,  $x^{(i)} \in \mathbb{R}^n$ , and  $b \in \mathbb{R}^M$

**Output:** Solution  $x^{(i+1)} \in \mathbb{R}^n$

```

// Initialization
 $\beta \leftarrow \|b - Ax^{(i)}\|_2$ ,  $u \leftarrow (b - Ax^{(i)})/\beta$ 
 $\alpha \leftarrow \|A^T u\|_2$ ,  $v \leftarrow A^T u/\alpha$ 
 $\bar{\rho} \leftarrow \beta$ ,  $\bar{\phi} \leftarrow \alpha$ 
while  $x$  is not converged do
  // Bidiagonalization
   $\beta \leftarrow \|Av - \alpha u\|_2$ ,  $u \leftarrow (Av - \alpha u)/\beta$ 
   $\alpha \leftarrow \|A^T u - \beta v\|_2$ ,  $v \leftarrow (A^T u - \beta v)/\alpha$ 
  // Construct and apply next orthogonal transformation
   $c \leftarrow \bar{\rho}/\sqrt{\bar{\rho}^2 + \beta^2}$ ,  $s \leftarrow \beta/\rho$ 
   $\theta \leftarrow s\alpha$ ,  $\phi \leftarrow c\bar{\phi}$ 
   $\bar{\rho} \leftarrow -c\alpha$ ,  $\bar{\phi} \leftarrow s\bar{\phi}$ 
  // Update the solution and weight
   $x \leftarrow x + w\phi/\rho$ 
   $w \leftarrow v - w\theta/\rho$ 
end while
 $x^{(i+1)} \leftarrow x$ 

```

---

## 4 Performance Evaluation

To evaluate the computational efficiency of the proposed method, we test the algorithm using the following weighted norms minimization problems:

$$\text{(Problem 1)} \quad \min_x \|A_1 x - b_1\|_1, \quad (15)$$

$$\text{(Problem 2)} \quad \min_x \|A_2 x - b_2\|_2^2 + \|A_3 x - b_3\|_1, \quad (16)$$

where  $A_1 \in \mathbb{R}^{500 \times 400}$ ,  $b_1 \in \mathbb{R}^{500}$  and  $A_2, A_3 \in \mathbb{R}^{1000 \times 800}$ ,  $b_2, b_3 \in \mathbb{R}^{1000}$ . Matrices  $A_k$  and solution  $x$  are randomly generated, and vector  $b_k$  is computed by  $b_k \leftarrow A_k x$ . To add outliers to the systems, the signs of 10% of elements in  $b_k$  are flipped.

We implement the following methods to compare the computational times on a PC equipped with Intel Core i7 930 @2.8GHz and 24 GB Memory.

- L-BFGS
- IRLS by QR decomposition with column pivoting (IRLS-QR)
- IRLS by Jacobi-preconditioned Conjugate Gradient method (IRLS-CG)
- IRLS by Jacobi-preconditioned Conjugate Gradient method with warm start (IRLS-CG-WS)
- IRLS by LSQR (IRLS-LSQR)
- IRLS by LSQR with warm start (IRLS-LSQR-WS) (proposed method)

We use ALGLIB [31] for L-BFGS and Eigen [32] matrix library for matrix operations. Because L-BFGS does not converge well on Problem 1 and Problem 2 that include non-smooth  $\ell_1$ -norms, we approximate  $\|A_1x - b_1\|_1$  to  $\sqrt{(A_1x - b_1)^2 + \gamma}$  with a sufficiently small positive value  $\gamma (= 10e^{-8})$  when applying L-BFGS.

Table 1 summarizes the average computation times and residuals of ten trials, and Figs. 2 and 3 show computation times and residuals over iterations when solving Problems 1 and 2, respectively. As shown in Table 1, IRLS-based methods are faster and more accurate than L-BFGS, even though we use a relaxed tolerance for L-BFGS for faster convergence. Although IRLS-QR solves the smaller problem fast with the smallest residual, the computation time rapidly grows as the size of the matrix becomes larger.

The effectiveness of the warm start strategy is clearly seen in IRLS by Conjugate Gradient and LSQR, showing about 20 times faster convergence. As shown in Fig. 2, while computation times for methods without warm start drastically increase as the iteration proceeds, they are significantly reduced with warm start. It is also shown in Fig. 3 that the warm start strategy is effective in reducing the residual by providing a guide to solve least-squares with high condition numbers at later iterations.

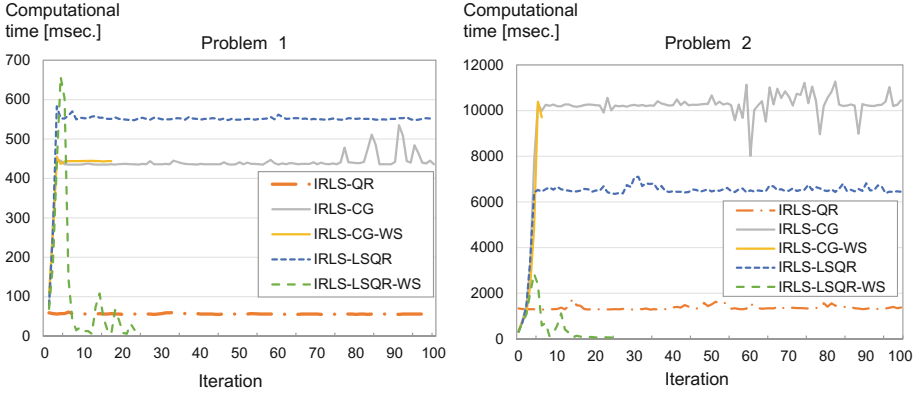
**Table 1.** Computation times and residuals of each method applied to problems 1 and 2.

	Problem 1		Problem 2	
	Computation time (sec.)	Residual	Computation time (sec.)	Residual
L-BFGS	710*	741	$> 1.0e^5$	–
IRLS-QR	5.6	642	137 <sup>†</sup>	2456 <sup>†</sup>
IRLS-CG	43	763	990 <sup>†</sup>	2468 <sup>†</sup>
IRLS-CG-WS	7.0	664	30	2468
IRLS-LSQR	54	721	634 <sup>†</sup>	2620 <sup>†</sup>
IRLS-LSQR-WS (ours)	2.6	682	14	2475

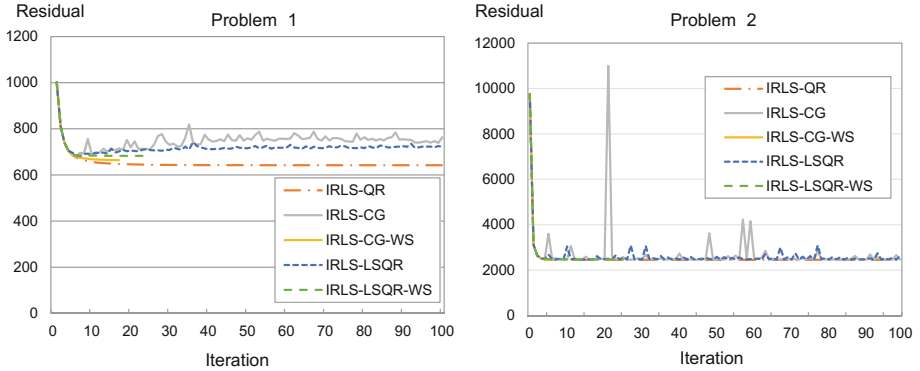
\*Due to the enormous computational time of L-BFGS, we set the tolerance greater ( $1.0e^{-6}$ ) than usual ( $1.0e^{-8}$ ).

<sup>†</sup>Because these methods did not converge, the computational times and residuals at 100-th iteration are shown.





**Fig. 2.** Variation of computation times over iterations. Our method benefits from the warm start strategy and the computation time quickly drops at the early stage of iterations.



**Fig. 3.** Variation of residuals over iterations.

## 5 Applications

The expression of the general norm approximation (4) offers flexibility of treating diverse objective functions in a unified manner, and they can generally benefit from the proposed efficient computation method. As example use cases, we show two applications in this section: Photometric stereo in Sect. 5.1 and surface reconstruction from normals in Sect. 5.2.

### 5.1 Photometric Stereo

Photometric stereo is a method for estimating surface normal of a surface from its appearance variations under different lightings. Let us assume that we have an  $f \times 3$  light direction matrix  $L$  that contains  $f$  distinct light directions as

row vectors. A scene is illuminated under these light directions, and it yields corresponding observation vector  $o \in \mathbb{R}_+^f$  for each pixel. Assuming that the scene reflectance obeys the Lambert's law, the image formation model of a pixel can be expressed as can be expressed as

$$o = Ln, \quad (17)$$

where  $n$  is a surface normal vector that we wish to estimate scaled by diffuse albedo. When the number of light directions is greater than three ( $f > 3$ ), the Lambertian photometric stereo [33] determines the scaled surface normal by the least-squares approximate solution as

$$n^* = \underset{n}{\operatorname{argmin}} \|Ln - o\|_2^2. \quad (18)$$

The solution method for the above problem is rather straightforward and does not even require our method, while it can still be represented as a special case of (4).

In reality, scene reflectances may contain specularity that can be regarded as unmodelled outliers. It motivated the previous work [34, 35] to use an  $\ell_1$ -norm minimization as robust estimation as

$$n^* = \underset{n}{\operatorname{argmin}} \|Ln - o\|_1. \quad (19)$$

It corresponds to minimization of the residual  $e = o - Ln$ , as above can be rewritten as the problem  $\min_n \|e\|_1$  s.t.  $e = o - Ln$ . This hard constraint can be relaxed as a regularizer as depicted in [34, 35] as

$$n^* = \underset{n}{\operatorname{argmin}} \lambda_1 \|o - Ln - e\|_2^2 + \|e\|_1 \quad (20)$$

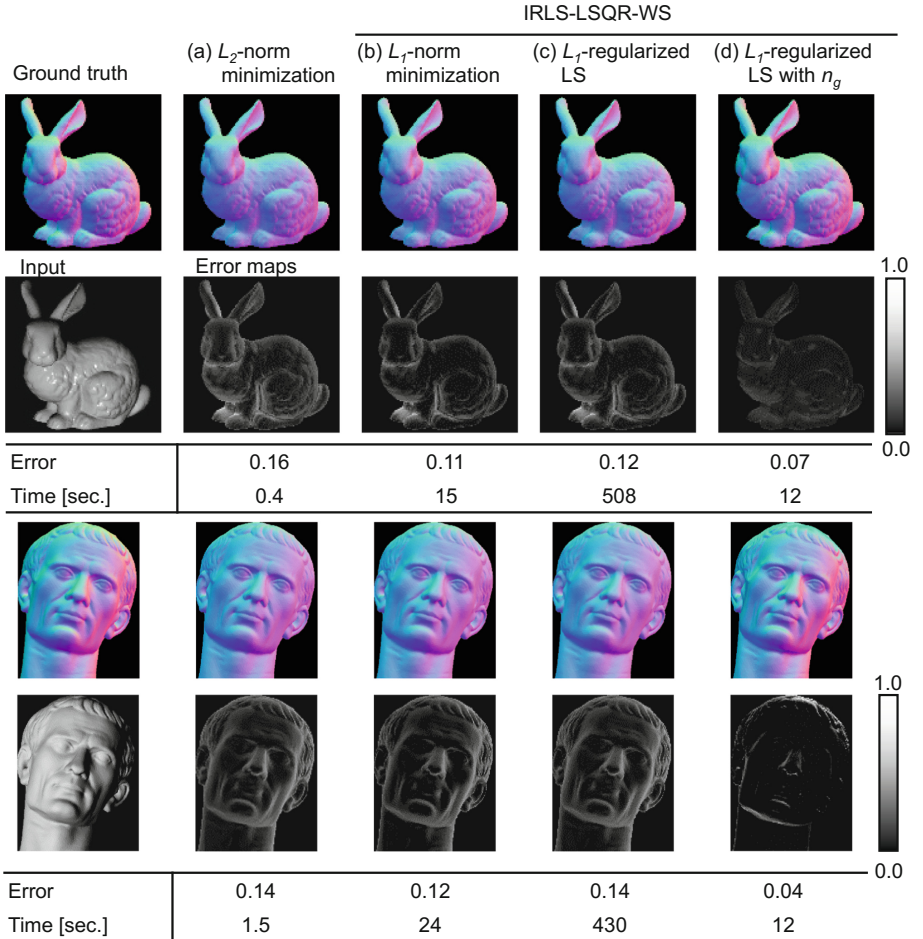
for allowing a certain magnitude of Gaussian error in the constraint.

In a special case, when we have an external method for determining surface normal, *e.g.*, surface normal obtained from a measured depth by structured light, and can use it as prior for stabilizing the solution, the photometric stereo problem can be formulated as

$$n^* = \underset{n}{\operatorname{argmin}} \|Ln - o\|_1 + \lambda_2 \|n - n_g\|_2^2, \quad (21)$$

in which  $n_g$  is surface normal that are obtained by an external method.

These four different problem settings (18)–(21) can all be represented by general norm approximation (4), and our method is applicable to any of the settings. To demonstrate this, we render 40 images of *Bunny* and *Caesar* scenes under different light directions and solve these four problems. We set  $\lambda_1, \lambda_2 = 10e^6$ .  $n_g$  in (21) is generated by adding 1% of Gaussian noise to every normal elements in the ground truth normal vector. Figure 4 shows estimated surface normal and error maps of the proposed IRLS-LSQR-WS method. Putting aside that there are variations in errors because of different formulations, it shows that our method is applicable to diverse formulations.



**Fig. 4.** Surface normal estimates and error maps with photometric stereo. (a)–(d) correspond to the settings (18)–(21). “Error” indicates the angular error in degrees, and “Time” corresponds to the computation time.

## 5.2 Surface Reconstruction from Normals

Once the surface normal is obtained by photometric stereo or shape from shading, one may like to reconstruct a surface from the normal, *e.g.*, by [36]. From surface normal  $n = (n_x, n_y, n_z)^T$  defined in the image coordinates  $(u, v)$ , the gradients  $g_u$  and  $g_v$  are computed as

$$g_x(u, v) = \frac{n_x(u, v)}{n_z(u, v)}, \quad g_y(u, v) = \frac{n_y(u, v)}{n_z(u, v)}. \quad (22)$$

Let  $G_u, G_v \in \mathbb{R}^{m \times n}$  denote matrices of  $g_u$  and  $g_v$ , respectively. These gradients corresponds to the 1-st order differentiation of the surface  $Z \in \mathbb{R}^{m \times n}$  to

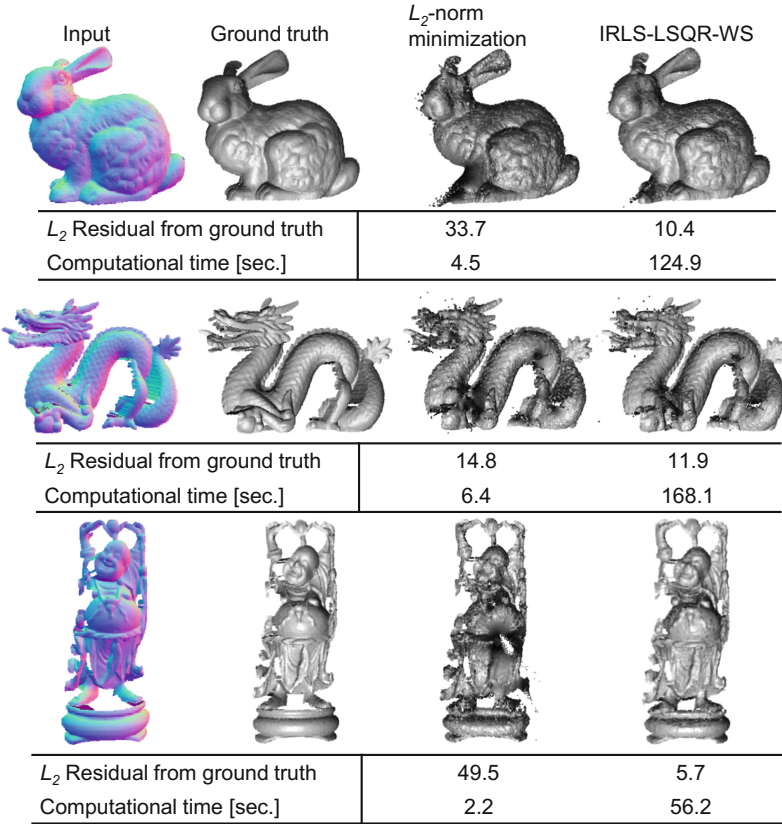
be reconstructed. Therefore, with a differentiation matrix  $D \in \mathbb{R}^{2mn \times mn}$ , the relationship can be written as

$$Dz \approx g, \tag{23}$$

where

$$D = \frac{1}{h} \begin{bmatrix} D_x \\ D_y \end{bmatrix}, \quad z = \text{vec}(Z), \quad g = \begin{bmatrix} \text{vec}(G_u) \\ \text{vec}(G_v) \end{bmatrix},$$

$$D_x = \begin{bmatrix} B & & & \\ & B & & \\ & & \ddots & \\ & & & B \end{bmatrix}, \quad D_y = \begin{bmatrix} -I & I & & & & \\ & -I & I & & & \\ & & & \ddots & \ddots & \\ & & & & -I & I \\ & & & & I & -I \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \\ & & & & 1 & -1 \end{bmatrix},$$



**Fig. 5.** Surface reconstruction from normal maps by the proposed method (IRLS-LSQR-WS)

with the identity matrix  $I$  and pixel size  $h$ . Similar to [37], we consider that there are outliers in normals that are 10 times larger than true normals and 10% of pixels are corrupted by them. In this setting, we consider the  $\ell_1$ -residual minimization to reconstruct the surface in the presence of outliers as

$$\min_z \|Dz - g\|_1. \quad (24)$$

This problem, again, is a special case of the general norm approximation problem (4), and our method can be applied to derive the solution  $z$ . We use three different scenes, *Bunny*, *Dragon*, and *Happy Buddha* as target scenes for testing this scenario. For comparison, we show the result of surface reconstruction by  $\ell_2$ -norm minimization as reference.

Figure 5 shows the reconstructed surfaces,  $\ell_2$ -residual from the ground truth, and computation times. The accuracy indicates the strength of  $\ell_1$ -residual minimization, but more importantly, our method is capable of handling any of these formulations because of the generalized form of norm minimization (4).

## 6 Conclusions

We presented a fast general norm approximation that can be applicable to diverse problem settings in computer vision. The proposed method (IRLS-LSQR-WS) is assessed in comparison to other state-of-the-art techniques and shows the favorable computation efficiency and accuracy at a time. In addition to the numerical tests, we show application scenarios by taking photometric stereo and surface reconstruction as examples to illustrate the usefulness of the general norm approximation.

During the experiments, we found that the proposed method is advantageous over IRLS-CG-WS in terms of stability, *i.e.*, IRLS-CG-WS occasionally becomes unstable when the condition number of the problem grows rapidly, while the proposed method does not suffer from this issue. On the other hand, IRLS-CG-WS tends to converge slightly faster when the design matrix  $A$  is sparse compared to the proposed method. We are interested in studying this trade-off by characterizing the problem by looking into the design matrix structure. In addition, further acceleration by preconditioning for LSQR [38] is another interesting venue to investigate.

**Acknowledgement.** This work was partly supported by JSPS KAKENHI Grant Numbers JP16H01732 and JP26540085.

## References

1. Lu, M., Zheng, B., Takamatsu, J., Nishino, K., Ikeuchi, K.: In: 3D Shape Restoration via Matrix Recovery. Springer, Heidelberg (2011)
2. Futragoon, N., Kitamoto, A., Andaroodi, E., Matini, M.R., Ono, K.: In: 3D Reconstruction of a Collapsed Historical Site from Sparse Set of Photographs and Photogrammetric Map. Springer, Heidelberg (2011)

3. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* **58**, 267–288 (1996)
4. Tikhonov, A.N., Arsenin, V.Y.: *Solution of Ill-posed Problems*. Winston & Sons, Washington (1977). ISBN 0-470-99124-0
5. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B* **67**, 301–320 (2005)
6. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49**, 409–436 (1952)
7. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**, 217–288 (2011)
8. Gentle, J.E.: In: *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer, New York (2007). ISBN 978-0-387-70872-0
9. Karmarkar, N.: A new polynomial-time algorithm for linear programming. *Combinatorica* **4**, 373–395 (1984)
10. Daubechies, I., DeVore, R., Fornasier, M., Gunturk, S.: Iteratively re-weighted least squares minimization: proof of faster than linear rate for sparse recovery. In: *42nd Annual Conference on Information Sciences and Systems*, pp. 26–29 (2008)
11. Aftab, K., Hartley, R.: Convergence of iteratively re-weighted least squares to robust m-estimators. In: *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 480–487 (2015)
12. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.* **8**, 43–71 (1982)
13. Lawson, C.L.: Contributions to the theory of linear least maximum approximations. Ph.D. thesis, UCLA (1961)
14. Rice, J.R., Usow, K.H.: The lawson algorithm and extensions. *Math. Comput.* **22**, 118–127 (1968)
15. Gorodnitsky, I.F., Rao, B.D.: Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.* **45**, 600–616 (1997)
16. Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3869–3872 (2008)
17. Candès, E.J., Wakin, M.B., Boyd, S.: Enhancing sparsity by reweighted  $\ell_1$  minimization. *J. Fourier Anal. Appl.* **14**, 877–905 (2008)
18. Wipf, D.P., Nagarajan, S.: Iterative reweighted  $\ell_1$  and  $\ell_2$  methods for finding sparse solutions. *J. Sel. Top. Signal Process* **4**(2), 317–329 (2010)
19. Burrus, C.S., Barreto, J., Selesnick, I.W.: Iterative reweighted least-squares design of fir filters. *IEEE Trans. Signal Process.* **42**, 2926–2936 (1994)
20. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.* **26**, 70 (2007). *Proceedings of SIGGRAPH*
21. Joshi, N., Zitnick, L., Szeliski, R., Kriegman, D.: Image deblurring and denoising using color priors. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009)
22. Wohlberg, B., Rodríguez, P.: An iteratively reweighted norm algorithm for minimization of total variation functionals. *IEEE Signal Process. Lett.* **14**, 948–951 (2007)
23. Liu, C., Sun, D.: On Bayesian adaptive video super resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**, 346–360 (2014)

24. Mohan, K., Fazel, M.: Iterative reweighted algorithms for matrix rank minimization. *J. Mach. Learn. Represent.* **13**, 3441–3473 (2012)
25. Chen, C., Huang, J., He, L., Li, H.: Preconditioning for accelerated iteratively reweighted least squares in structured sparsity reconstruction. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2713–2720 (2014)
26. Fornasier, M., Peter, S., Rauhut, H., Worm, S.: Conjugate gradient acceleration of iteratively re-weighted least squares methods. *Comput. Optim. Appl.* **65**, 205–259 (2016)
27. Shewchuk, J.R.: An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA (1994)
28. Howell, G.W., Baboulin, M.: LU preconditioning for overdetermined sparse least squares problems. In: Wyrzykowski, R., Deelman, E., Dongarra, J., Karczewski, K., Kitowski, J., Wiatr, K. (eds.) *PPAM 2015. LNCS*, vol. 9573, pp. 128–137. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-32149-3\\_13](https://doi.org/10.1007/978-3-319-32149-3_13)
29. Benbow, S.J.: Solving generalized least-squares problems with LSQR. *SIAM J. Matrix Anal. Appl.* **21**, 166–177 (1999)
30. Nolet, G.: Solving Large Linearized Tomographic Problems: Seismic Tomography, Theory and Practice, pp. 227–247. Chapman and Hall, London (1993)
31. Bochkhanov, S., Bystriitsky, V.: ALGLIB. <http://www.alglib.net/>
32. Guennebaud, G., Jacob, B., et al.: Eigen v3
33. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. *Opt. Eng.* **19**, 191139–191139 (1980)
34. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y., Ma, Y.: Robust photometric stereo via low-rank matrix completion and recovery. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) *ACCV 2010. LNCS*, vol. 6494, pp. 703–717. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-19318-7\\_55](https://doi.org/10.1007/978-3-642-19318-7_55)
35. Ikehata, S., Wipf, D., Matsushita, Y., Aizawa, K.: Robust photometric stereo via low-rank matrix completion and recovery. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012)
36. Harker, M., O’leary, P.: Regularized reconstruction of a surface from its measured gradient field. *J. Math. Imaging Vis.* **51**, 46–70 (2015)
37. Reddy, D., Agrawal, A.K., Chellappa, R.: Enforcing integrability by error correction using l1-minimization. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2350–2357 (2009)
38. Avron, H., Maymounkov, P., Toledo, S.: Blendenpik: supercharging lapack’s least-squares solver. *SIAM J. Sci. Comput.* **32**, 1217–1236 (2010)