

## Research Article

# Sparse-Representation-Based Direct Minimum $L^P$ -Norm Algorithm for MRI Phase Unwrapping

Wei He,<sup>1</sup> Ling Xia,<sup>1</sup> and Feng Liu<sup>2</sup>

<sup>1</sup> Department of Biomedical Engineering, Zhejiang University, Hangzhou 310027, China

<sup>2</sup> School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, QLD 4072, Australia

Correspondence should be addressed to Ling Xia; xialing@zju.edu.cn

Received 6 December 2013; Revised 8 February 2014; Accepted 12 February 2014; Published 26 March 2014

Academic Editor: Xiaobo Qu

Copyright © 2014 Wei He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A sparse-representation-based direct minimum  $L^P$ -norm algorithm is proposed for a two-dimensional MRI phase unwrapping. First, the algorithm converts the weighted- $L^P$ -norm-minimization-based phase unwrapping problem into a linear system problem whose system (coefficient) matrix is a large, symmetric one. Then, the coefficient-matrix is represented in the sparse structure. Finally, standard direct solvers are employed to solve this linear system. Several wrapped phase datasets, including simulated and MR data, were used to evaluate this algorithm's performance. The results demonstrated that the proposed algorithm for unwrapping MRI phase data is reliable and robust.

## 1. Introduction

From the MRI complex data, the phase information can be extracted with a restricted interval  $(-\pi, \pi]$ . That is, the phase value is wrapped. We call it a wrapped phase or the principal value  $\psi$ . This relationship between the wrapped phase and its corresponding true phase  $\theta$  can be described by  $\psi = \omega\{\theta\} = \theta + 2k\pi$ , where  $k$  is an integer and  $\omega\{\cdot\}$  is the wrapping operator, forcing the value of its argument inside the curly braces into the range  $(-\pi, \pi]$  by adding or subtracting an integral multiple of  $2\pi$  radians from its argument. However, what is needed is the true unknown phase  $\theta$ , because this relates to certain properties of interest, such as the velocity of the moving spins, the main  $B_0$  field inhomogeneity, and the magnetic susceptibility variations. Given the wrapped phase, phase unwrapping is applied to restore the true phase, obtaining the unwrapped estimate  $\varphi$ . This technique is an important tool in many MRI applications, for example, three-point Dixon water and fat separation [1], MR venography [2, 3], motion tracking in a tagged cardiac MRI [4], and field mapping in EPI [5].

Should the wrapped phases have no inconsistencies, the process of the phase unwrapping will be merely integrating the phase gradients over a path that covers the whole domain

of interest. This process is quite simple and path independent. No matter which path is followed, the results will be the same regardless of the constant offset. However, in practice, there are always inconsistencies owing to the presence of the noise, undersampling, and/or object discontinuities. Consequently, phase unwrapping becomes intractable and path dependent. The inconsistency is usually called “residue” and is detected by summing the wrapped phase gradients around each  $2 \times 2$  closed loop in the two-dimensional (2D) array, as shown in Figure 1. Consider

$$\Delta\psi_{i,j}^x - \Delta\psi_{i,j}^y - \Delta\psi_{i,j+1}^x + \Delta\psi_{i+1,j}^y = \pm 2\pi, \quad (1)$$

where  $\Delta\psi_{i,j}^x$  and  $\Delta\psi_{i,j}^y$  are defined to be the gradients of the wrapped phases:

$$\begin{aligned} \Delta\psi_{i,j}^x &= \omega\{\psi_{i+1,j} - \psi_{i,j}\}, \\ \Delta\psi_{i,j}^y &= \omega\{\psi_{i,j+1} - \psi_{i,j}\}. \end{aligned} \quad (2)$$

In the literature, numerous two-dimensional (2D) phase unwrapping approaches have been developed. They can be classified into four categories: path-following [6–11], minimum-norm [12–18], Bayesian/regularization [19, 20], and

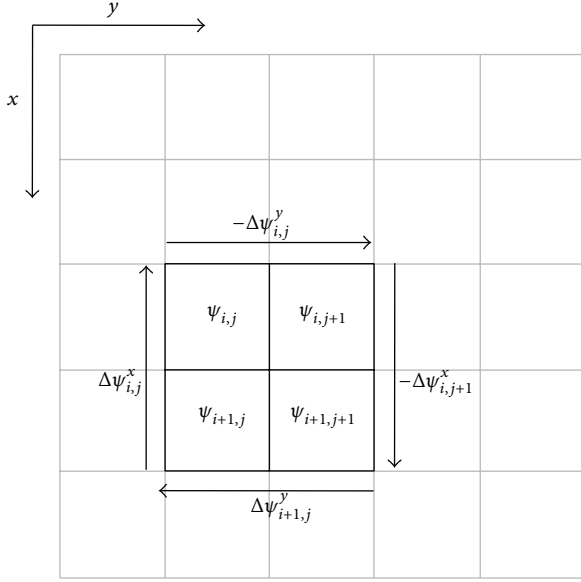


FIGURE 1: Residue calculation.

parametric modeling [21] methods. Our method belongs to the second category; thus, we briefly introduce the minimum-norm methods.

The minimum-norm methods estimate the unwrapped phases by minimizing the  $L^p$ -norm of the differences between the gradients of the wrapped and the unwrapped phases. With  $p = 2$ , this results in least-square algorithms. These employ the FFT/DCT transforms [18, 22] or/and iterative techniques [22] to reach an approximation for the least-square solution. The exact least-square solution is obtained by applying network programming techniques in [15]. Nevertheless, the least-square minimization tends to smooth the discontinuities, unless these discontinuities are given in advance as binary weights. The minimum  $L^1$ -norm algorithms [13, 23] have a better ability than the  $L^2$ -norm ones to preserve the discontinuities. As  $p$  approaches zero, the minimum  $L^p$ -norm method tends to obtain a more reliable result [24, 25]. Thus, the  $L^0$ -norm minimization is widely accepted as the most desirable in practice. However, the  $L^0$ -norm minimization is a nondeterministic polynomial-time hard (NP-hard) problem [14] with only approximate, not exact, solutions being developed in [12, 14]. The conventional minimum  $L^p$ -norm method [12] converts the  $L^p$ -norm minimization problem into a generalized matrix equation with a flexible option of  $p$ . It yields more accurate solutions than other methods, except for the cases including some residues closer to the periphery than to the other residues with opposite polarity. Additionally, because it is implemented in a dual-iterative structure (an iteration structure embedded in an outer one), it can be computationally intense.

In this work, a sparse-representation-based direct minimum  $L^p$ -norm (SDML <sup>$P$</sup> ) algorithm is proposed. It introduces the user-defined weights into the generalized matrix equation of the conventional minimum  $L^p$ -norm method to improve the performance of phase unwrapping, because, in this way,

the discontinuities in the unwrapped phase surface can be confined to the low-quality or zero-weight regions [22]. On the other hand, the sparse representations of the matrices in the modified matrix equation and the direct solvers are exploited in this algorithm to significantly reduce the computational time of the conventional minimum  $L^p$ -norm method for MRI phase unwrapping.

## 2. Materials and Methods

**2.1. Mathematics Foundation.** In general, for an  $M \times N$  2D wrapped phase array, the weighted minimum  $L^p$ -norm phase unwrapping problem is expressed by [14, 17]

$$\begin{aligned} \operatorname{argmin}_{\varphi} \sum_{i=1}^{M-1} \sum_{j=1}^N w_{i,j}^x \cdot |\varphi_{i+1,j} - \varphi_{i,j} - \Delta\psi_{i,j}^x|^p \\ + \sum_{i=1}^M \sum_{j=1}^{N-1} w_{i,j}^y \cdot |\varphi_{i,j+1} - \varphi_{i,j} - \Delta\psi_{i,j}^y|^p, \end{aligned} \quad (3)$$

where  $w_{i,j}^x$  and  $w_{i,j}^y$  are the user-defined weights for corresponding differences and indicate where the phase values are more reliable than others. The user-defined weights are given by

$$\begin{aligned} w_{i,j}^x &= \min(w_{i+1,j}, w_{i,j}), \\ i &= 1, 2, \dots, M-1; \quad j = 1, 2, \dots, N, \\ w_{i,j}^y &= \min(w_{i,j+1}, w_{i,j}), \\ i &= 1, 2, \dots, M; \quad j = 1, 2, \dots, N-1, \end{aligned} \quad (4)$$

where

$$\begin{aligned} w_{i,j} &= \frac{\max(z_{i,j}) - z_{i,j}}{\max(z_{i,j}) - \min(z_{i,j})}, \\ z_{i,j} &= \frac{\sqrt{\sum (\Delta\psi_{i,j}^x - \overline{\Delta\psi_{m,n}^x})^2} + \sqrt{\sum (\Delta\psi_{i,j}^y - \overline{\Delta\psi_{m,n}^y})^2}}{l^2}. \end{aligned} \quad (5)$$

For each sum, the indexes  $(i, j)$  cover over the  $l \times l$  window centered at the pixel  $(m, n)$ . The terms  $\Delta\psi_{i,j}^x$  and  $\Delta\psi_{i,j}^y$  are the wrapped phase gradients in the  $l \times l$  windows and  $\overline{\Delta\psi_{m,n}^x}$  and  $\overline{\Delta\psi_{m,n}^y}$  are the averages of these wrapped phase gradients. In this paper  $l = 3$ .  $w_{i,j}$  is relatively higher in the areas where the phase changes smoothly.

Through the analogous derivation process in [12, 26], the minimum  $L^p$ -norm solution of (3) eventually entails the solution of the following linear system of equations:

$$\begin{aligned} R_{i,j}(\varphi_{i+1,j} - \varphi_{i,j}) + C_{i,j}(\varphi_{i,j+1} - \varphi_{i,j}) \\ - R_{i,j}(\varphi_{i,j} - \varphi_{i-1,j}) - C_{i,j}(\varphi_{i,j} - \varphi_{i,j-1}) = \rho_{i,j} \end{aligned} \quad (6)$$

where

$$R_{i,j} = w_{i,j}^x \cdot \left| \varphi_{i+1,j} - \varphi_{i,j} - \Delta\psi_{i,j}^x \right|^{p-2}, \quad (7)$$

$$i = 1, 2, \dots, M-1; \quad j = 1, 2, \dots, N,$$

$$C_{i,j} = w_{i,j}^y \cdot \left| \varphi_{i,j+1} - \varphi_{i,j} - \Delta\psi_{i,j}^y \right|^{p-2}, \quad (8)$$

$$i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N-1,$$

$$\rho_{i,j} = R_{i,j}\Delta\psi_{i,j}^x - R_{i,j}\Delta\psi_{i-1,j}^x + C_{i,j}\Delta\psi_{i,j}^y - C_{i,j}\Delta\psi_{i,j-1}^y. \quad (9)$$

That is to say, when the weighted minimum  $L^p$ -norm phase unwrapping solution is desired, it can be simpler to find the solution  $\varphi_{i,j}$  of (6) instead.

For the setting of the values of data-dependent weights  $R_{i,j}$  and  $C_{i,j}$  empirically, the weights are always normalized in the interval  $[0, 1]$  in practice. Hence, (7) and (8) are modified to incorporate this constraint. To avoid  $|\cdot|^{p-2}$  with  $0 \leq p \leq 2$  being extremely great or even infinite, the normalization is defined by

$$R_{i,j} = w_{i,j}^x \cdot \frac{\alpha}{\left| \varphi_{i+1,j} - \varphi_{i,j} - \Delta\psi_{i,j}^x \right|^{2-p} + \alpha},$$

$$i = 1, 2, \dots, M-1; \quad j = 1, 2, \dots, N, \quad (10)$$

$$C_{i,j} = w_{i,j}^y \cdot \frac{\alpha}{\left| \varphi_{i,j+1} - \varphi_{i,j} - \Delta\psi_{i,j}^y \right|^{2-p} + \alpha},$$

$$i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N-1.$$

Here, setting  $\alpha$  to be 0.01 radians is a compromise between accuracy and efficiency [12, 22].

## 2.2. Sparse-Representation-Based Direct Minimum $L^p$ -Norm (SDML<sup>P</sup>) Algorithm

**2.2.1. Sparse Representation of the Objective.** The estimated phase distribution is shown in Figure 1. To simplify the algorithm, concatenating the columns of the phase matrix yields a vector of length  $MN$ :

$$\phi = [\varphi_{1,1}, \varphi_{2,1}, \dots, \varphi_{M,1}, \varphi_{1,2}, \varphi_{2,2}, \dots, \varphi_{M,2}, \dots, \varphi_{1,N}, \dots, \varphi_{M,N}]^T. \quad (11)$$

The superscript  $T$  refers to a matrix transpose.

The wrapped phase differences along the two directions,  $\Delta\psi^x$  and  $\Delta\psi^y$ , are also matrices. Their sizes are  $(M-1) \times N$  and  $M \times (N-1)$ , respectively. As above, we concatenate the columns of each matrix and then merge these two vectors vertically into a single array:

$$d = [\Delta\psi_{1,1}^x, \dots, \Delta\psi_{M-1,1}^x, \dots, \Delta\psi_{1,N}^x, \dots, \Delta\psi_{M-1,N}^x, \Delta\psi_{1,1}^y, \dots, \Delta\psi_{M,1}^y, \dots, \Delta\psi_{1,N-1}^y, \dots, \Delta\psi_{M,N-1}^y]^T. \quad (12)$$

The length of  $d$  is  $(M-1)N + M(N-1)$ .

Analogous to the matrix equation used in the weighted least-squares phase unwrapping algorithm [18], the system of equations defined by (6) can be represented in matrix form as follows:

$$\mathbf{Q}\phi = s, \quad (13)$$

where  $\mathbf{Q}$  denotes an  $MN \times MN$  matrix given by

$$\mathbf{Q} = \mathbf{A}^T \mathbf{W}^T \mathbf{W} \mathbf{A}, \quad (14)$$

$$s = \mathbf{A}^T \mathbf{W}^T \mathbf{W} d. \quad (15)$$

$\mathbf{A}$  is a matrix consisting of an  $(M-1)N \times MN$  upper partition and an  $M(N-1) \times MN$  lower partition:

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{D}_1 \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & -\mathbf{I} \end{bmatrix}, \quad (16)$$

where  $\mathbf{D}_1$  is an  $(M-1) \times M$  matrix given by

$$\mathbf{D}_1 = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix} \quad (17)$$

and  $\mathbf{I}$  is an  $M \times M$  identity matrix.

To make the expansion of (13) equal to (6), the matrix  $\mathbf{W}^T \mathbf{W}$  is constructed as

$$\mathbf{W}^T \mathbf{W} = \text{diag} \{ R_{1,1}, \dots, R_{M-1,1}, \dots, R_{1,N}, \dots, R_{M-1,N}, C_{1,1}, \dots, C_{M,1}, \dots, C_{1,N-1}, \dots, C_{M,N-1} \}, \quad (18)$$

where  $\text{diag}\{\cdot\}$  puts its arguments in order on the main diagonal. It is easy to see that  $\mathbf{W}^T \mathbf{W}$  is an  $[(M-1) \times N + M \times (N-1)]^2$  matrix.

If these foregoing matrices are stored and operated using a standard matrix structure, this will consume a significant amount of memory and become unfeasible in standard computers. For instance, given a  $256 \times 256$  phase image, the sizes of  $\mathbf{A}$ ,  $\mathbf{W}^T \mathbf{W}$ , and  $\mathbf{Q}$  are  $130560 \times 65536$ ,  $130560 \times 130560$ , and  $65536 \times 65536$ , respectively. These matrices are obviously too large (exceeding  $2^{30}$ ) to be stored and manipulated in a program. Fortunately, all these matrices are sparse and can be represented with a sparse structure. We store only nonzero entries of the matrix together with their indexes. That is, a 3-tuple,  $(i, j, a_{ij})$ , is applied to uniquely identify a nonzero entry of the sparse matrix [27, 28], where  $i$  and  $j$  are the row

and column indexes, respectively, and  $a_{ij}$  denotes the value of the nonzero entry located in  $(i, j)$ . Moreover, to make the subsequent computation more efficient, the nonzero entries are ordered first by columns and then by rows. Finally, the size of the original sparse matrix should also be stored.

It should be noted that, henceforth,  $\mathbf{A}$ ,  $\mathbf{W}^T\mathbf{W}$ , and  $\mathbf{Q}$  will still refer to the original sparse matrices but will be stored and operated, respectively, in the corresponding sparse structures.

**2.2.2. Direct Solver.** Equation (14) implies that  $\mathbf{Q}$  is a large sparse, real, symmetric matrix. Thus, (13) becomes a linear system of equations involving the large sparse symmetric coefficient matrix  $\mathbf{Q}$ . To solve this system of equations, considerable effort has been devoted over the past three decades [29]. Among these algorithms, the direct solvers that rely on the explicit factorization of the coefficient matrix are widely used owing to their generality and robustness. Especially for a symmetric coefficient matrix, the direct solvers prefer to adopt the  $LDL^T$  factorization [30], a variant of Gaussian elimination that can also be considered as an alternative form of the Cholesky factorization without extracting the square roots [31]. Under this factorization, the matrix  $\mathbf{Q}$  is decomposed into a lower triangular matrix  $\mathbf{L}$ , a diagonal matrix  $\mathbf{D}$  with  $1 \times 1$  and  $2 \times 2$  blocks, and the conjugate transpose of  $\mathbf{L}$ . Consider

$$\mathbf{Q} = \mathbf{LDL}^T. \quad (19)$$

Then, given the right-hand side  $s$ , the estimated phases in (13) can be obtained through a forward elimination followed by the backward substitution [28]. The time complexity is  $O((MN)^{1.5})$  and the memory complexity is  $O(MN \log(MN))$ . For more details of the direct solvers, please see [32].

**2.2.3. Implementation of the Algorithm.** During the last two decades a number of software packages that implement direct solvers have been developed [33]. If the coefficient matrix is positive definite, CHOLMOD [34] is adopted, because of its rapid computation and relatively small amounts of memory demand. If not, MA57 [35] is strongly recommended.

To accelerate the convergence, the termination condition is set to no residues for the distinctions [12]. First, the distinction is defined by

$$E_{i,j} = \omega \{ \psi_{i,j} - \varphi_{i,j} \}, \quad i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N. \quad (20)$$

Then, we treat these distinctions similarly to the phase data. Based on the theory that any wrapped phase image can be uniquely unwrapped (with an arbitrary constant offset) if it does not have residues [36], we check whether the  $E_{i,j}$  have residues. Given no residues, we unwrap  $E_{i,j}$  by a flood-fill algorithm [37, 38] with a centre start point and no branch cuts.

*Step 1.* Choose the start pixel as the point in the location of  $i = \text{round}(M/2)$  and  $j = \text{round}(N/2)$ , where  $\text{round}(\cdot)$  rounds its argument to the nearest integer. Its phase value is stored as an unwrapped phase value in the solution matrix.

The four neighbouring pixels are next unwrapped and their unwrapped phase values are placed in the solution matrix. These four pixels are inserted in the unwrapped list.

*Step 2.* Pick a pixel from the unwrapped list and then eliminate it from the unwrapped list. Unwrap the phase values of its four neighbouring pixels, avoiding pixels that have been unwrapped. Insert these pixels in the unwrapped list and put their unwrapped phase values in the solution matrix.

*Step 3.* Repeat Step 2 until the unwrapped list becomes empty.

Unwrapping  $E_{i,j}$  is signified by  $\omega^{-1}\{E_{i,j}\}$ , which is added back to the estimated phase to obtain the final result. Consider

$$\varphi_{i,j}^{\text{final}} = \varphi_{i,j} + \omega^{-1}\{E_{i,j}\}, \quad i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N. \quad (21)$$

In summary, the detailed procedure of the  $SDML^P$  phase unwrapping algorithm is described as below.

*Step 1.* Set the iteration time  $k = 0$ . Set the value of  $p$ . Initialize the estimated phases.

*Step 2.* Compute the weights  $R_{i,j}$  and  $C_{i,j}$  from (10).

*Step 3.* Compute  $\mathbf{Q}$  and  $s$  from (14) and (15); then solve (13) by the direct solver.

*Step 4.* Compute the distinction  $E_{i,j}$  by (20). Check  $E_{i,j}$  for residues. If there are no residues, continue. If  $k > k_{\max}$ , end. Otherwise, set  $k = k + 1$  and go to Step 2.

*Step 5.* Unwrap  $E_{i,j}$  by the flood-fill algorithm, and form the final estimated phases by (21).

**2.2.4. Evaluation.** The following weighted  $L^0$  measure is used to evaluate the quality of an unwrapped solution:

$$\rho = \frac{1}{MN} \left[ \sum_{i=1}^{M-1} \sum_{j=1}^N w_{i,j}^x |\varphi_{i+1,j} - \varphi_{i,j} - \Delta\psi_{i,j}^x|^0 + \sum_{i=1}^M \sum_{j=1}^{N-1} w_{i,j}^y |\varphi_{i,j+1} - \varphi_{i,j} - \Delta\psi_{i,j}^y|^0 \right]. \quad (22)$$

This counts the ratio of pixels where the gradients of the unwrapped solution mismatch the wrapped phase gradients, which is what we have also minimized with  $p = 0$ . Therefore, the lower weighted  $L^0$  measure indicates the better performance.

### 3. Results and Discussion

Several 2D datasets are used to evaluate the performance of the proposed  $SDML^P$  algorithm. The actual number of the iterations of the  $SDML^P$  algorithm varies, depending on the loops after which the distinctions of each example become

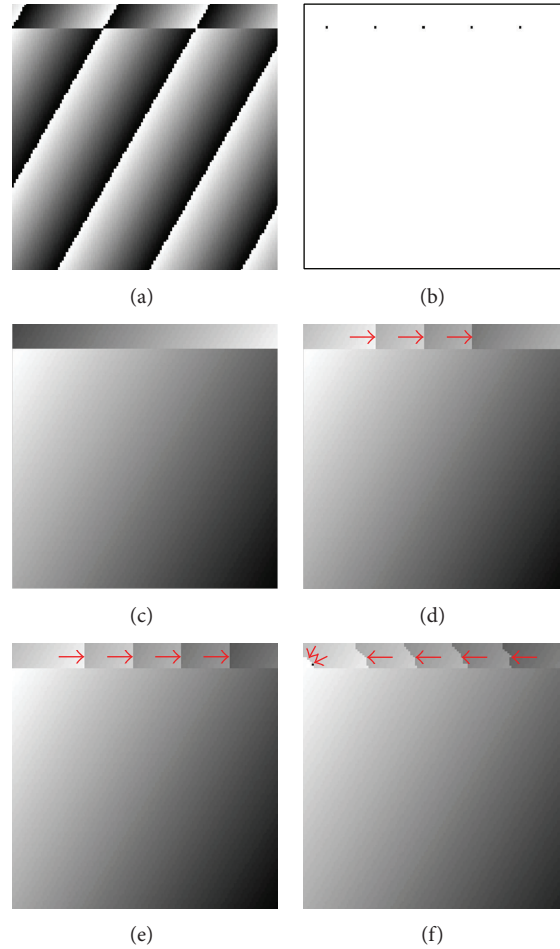


FIGURE 2: A  $128 \times 128$  phase dataset with shear, (a) its wrapped phase, and (b) its residue distribution including 6 residues. Unwrapped phase image using the (c)  $SDML^P$ , (d) conventional minimum  $L^p$ -norm, (e) PUMA, and (f) PHUN methods. Unwrapping errors are marked with red arrows.

residue free. The limited maximum number of iterations  $k_{\max}$  is set to be 20.

The performance of the proposed algorithm is compared with conventional minimum  $L^p$ -norm algorithm [12] and other two widely used 2D algorithms, PUMA [16] and PHUN [10]. Note that we choose  $p = 0$  for the first two methods, because  $L^0$ -norm minimization is more well behaved and most desired in practice (explained in Section 1). We merely display the best result we can obtain for the PHUN method in each example, because the behaviour of this algorithm is controlled by many parameters and the optimum values of the parameters vary for each dataset. All these methods are implemented in MATLAB (MathWorks, Natick, MA) on a PC (Intel 2 Quad CPU 2.39 GHz).

**3.1. Simulated Data.** We begin with a  $128 \times 128$  phase dataset with one shear line located horizontally along the 13th row below the upper border. The wrapped phase image is shown in Figure 2(a). Figure 2(b), where the black border is added, shows its residues marked as black dots.

The proposed method,  $SDML^P$ , made no unwrapping errors for this case, producing a solution (see Figure 2(c))

exactly the same as the true phase. However, while it obtained the smallest weighted  $L^0$  measure for this example, it converged more slowly than the PUMA and PHUN methods (see Figure 6). The conventional minimum  $L^p$ -norm and PUMA methods both yielded extra short vertical shear lines marked with red arrows in the upper parts of their results (see Figures 2(d) and 2(e)). Figure 2(f) shows the solution achieved by the PHUN method with five unexpected crooked shear lines and a black spot of outliers whose values are very low (all marked with red arrows).

Figure 3(a) shows a  $257 \times 257$  wrapped phase shear image with Gaussian noise (the signal-to-noise ratio is 0.8379 dB). Its residue distribution is shown in Figure 3(b).

The result of the  $SDML^P$  algorithm in Figure 3(c) is composed of a two-planar surface tearing along the median shear line as expected, except for some protrusions on this line where the phases are severely corrupted by both the noise and the object discontinuities. The conventional minimum  $L^p$ -norm method offers a slightly worse result, shown in Figure 3(d), where minor errors (marked with red arrows) occurred near the top and bottom of the median shear line. So the weighted  $L^0$  measure values of the  $SDML^P$  and



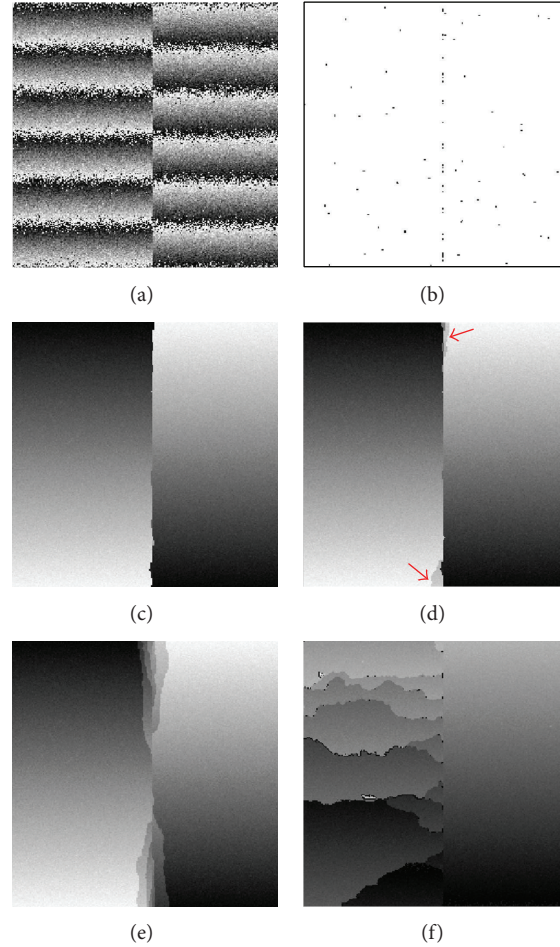


FIGURE 3: A  $257 \times 257$  noisy phase dataset with shear, (a) its wrapped phase, and (b) its residue distribution including 150 residues. Unwrapped phase image using the (c)  $SDML^P$ , (d) conventional minimum  $L^p$ -norm, (e) PUMA, and (f) PHUN methods.

conventional minimum  $L^p$ -norm methods are almost the same (see Figure 6(a)). However, it is noted in this case that the former converged much faster than the latter. The unwrapped phase image of the PUMA method in Figure 3(e) has some small-scale, anomalous “layer” artifacts around the median shear line. The PHUN method generated an incorrect unwrapped phase image with the left half full of “layer” artifacts that propagated from the median line to the surrounding regions or even the image border.

In summary, these two simulation examples, both having object discontinuities lying in the shear line, demonstrate the discontinuity-preserving ability of the  $SDML^P$  algorithm. Additionally, the  $SDML^P$  algorithm did not produce extra-undesired discontinuities that appeared as shear lines or “layer” artifacts.

**3.2. MR Data.** Figures 4(a) and 4(b) show the magnitudes and phases of a  $44 \times 44$  displacement encoded MR heart dataset [39], respectively.

The results of the  $SDML^P$ , the conventional minimum  $L^p$ -norm, and the PUMA algorithms are shown in Figures

4(d), 4(e), and 4(f). There is no significant visible difference between these three images. In these unwrapped phase images, the rough shape of the heart (compared to the magnitude picture in Figure 4(a)) is dimly visible. We then examine the corresponding discontinuity maps (added black borders) that are the distributions of the discontinuities where pixels (marked in black) differ from a neighbouring pixel by more than  $\pi$  radians. These maps in Figures 4(h), 4(i), and 4(j) have little differences. In addition, the weighted  $L^0$  measures, depicted in Figure 6(a), of these three methods are similar. The extremely fast algorithm, PHUN, offered a solution involving many black spots (see Figure 4(g)) that correspond roughly to the locations of the residues (see Figure 4(c)). Also, it created a large number of undesired discontinuities (see Figure 5(j)).

An MR head example [12] is shown in Figure 5. The  $SDML^P$ , the conventional minimum  $L^p$ -norm, and the PUMA algorithms all produced plausible unwrapped phase images, shown in Figures 5(c), 5(d), and 5(e). Moreover, the discontinuity maps (see Figures 5(g), 5(h), and 5(i)) and weighted  $L^0$  measures (see Figure 6(a)) of these three

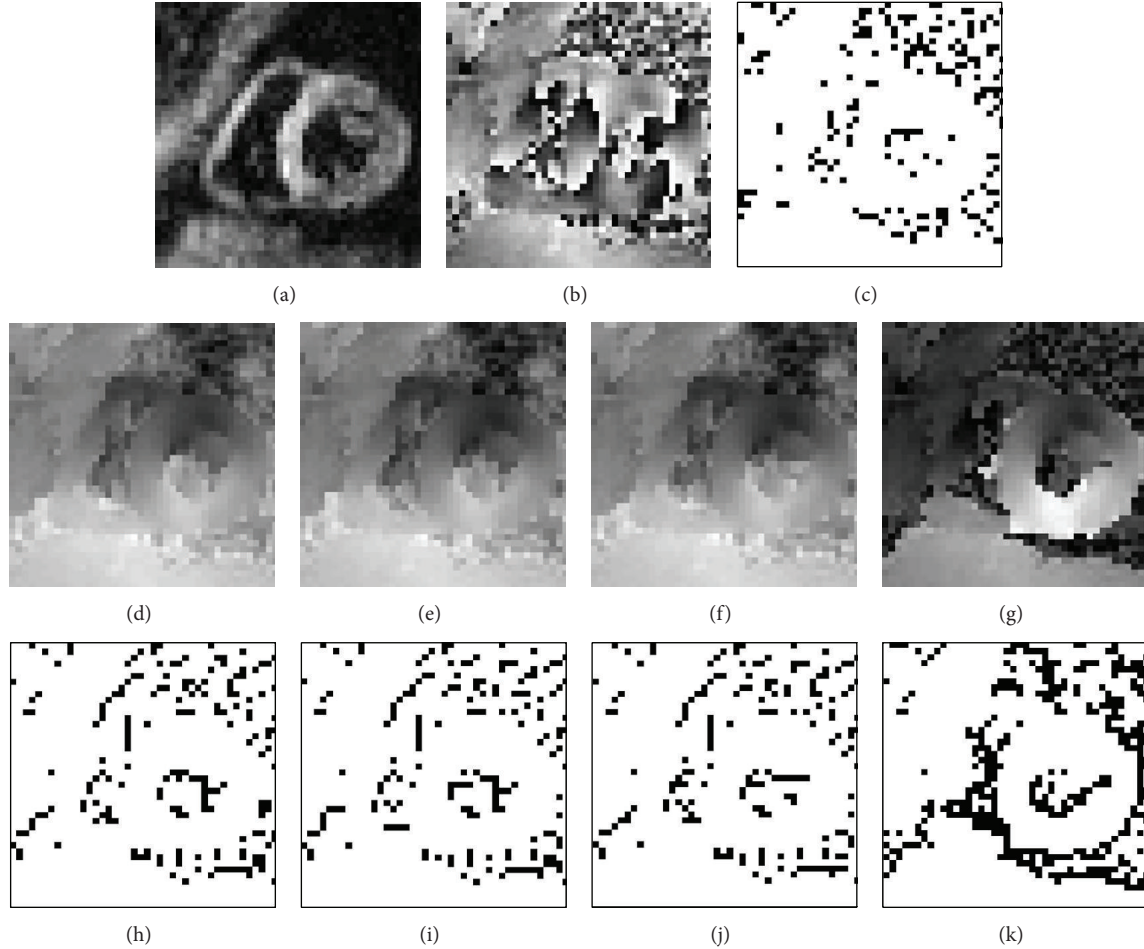


FIGURE 4: A  $44 \times 44$  displacement encoded MR heart dataset, (a) its magnitudes, (b) its wrapped phases, and (c) its residue distribution including 210 residues. Unwrapped phase image using the (d)  $SDML^P$ , (e) conventional minimum  $L^P$ -norm, (f) PUMA, and (g) PHUN methods. Discontinuities in the corresponding unwrapped phase images got by the (h)  $SDML^P$ , (i) conventional minimum  $L^P$ -norm, (j) PUMA, and (k) PHUN methods.

methods are almost the same. The PHUN method failed to unwrap this dataset correctly (see Figure 5(f)). Its result has a large number of undesired discontinuities (see Figure 5(j)).

The weighted  $L^0$  measures and execute time of all methods in the preceding four examples are compared in Figures 6(a) and 6(b), respectively. For a better visual effect, both of the ordinate axes are in a logarithmic scale. The  $SDML^P$  algorithm achieved the smallest weighted  $L^0$  measures for all four examples above. It significantly reduced the computational time compared to the conventional minimum  $L^P$ -norm method. However, the algorithm did not converge fast enough compared with the PUMA and PHUN algorithms. From the horizontal comparison of the execute time, it can be concluded that the execute time of the  $SDML^P$  algorithm depended partly on the size of the phase dataset. For example, the  $SDML^P$  algorithm converged in 0.5 seconds for the  $44 \times 44$  heart dataset, while, for the  $257 \times 257$  simulated dataset, it took 31 seconds.

The final example is the transverse section of a 5-slice MR knee dataset, as shown in Figure 7. The size of each slice is

$256 \times 256$ . The measurements were made with a 0.5T MRI scanner (Ningbo Xingyao Co., LTD., China).

Excellent results with no phase errors, shown in the third row of Figure 7, are derived from the unwrapping by the  $SDML^P$  algorithm. The conventional minimum  $L^P$ -norm algorithm worked well, except for generating two undesirable shear lines in the fourth slice. The PUMA algorithm yielded undesired results in the third and fourth slices with some shear lines but successfully unwrapped the other slices. The PHUN method made some unwrapping errors in all slices, shear lines in the first, third, and fourth slices, and white patches of outliers, whose values are very high, in the second and fifth slices. All the truncations and outliers are marked with yellow and green arrows, respectively, in Figure 7.

As above, the weighted  $L^0$  measures and execute time of all the methods for this multislice dataset are compared in Figures 8(a) and 8(b), respectively. The  $SDML^P$  method returned the smallest weighted  $L^0$  measures for all slices. Additionally, it yielded these results in  $19.62 \pm 0.99$  seconds, slower than the PUMA and PHUN methods.

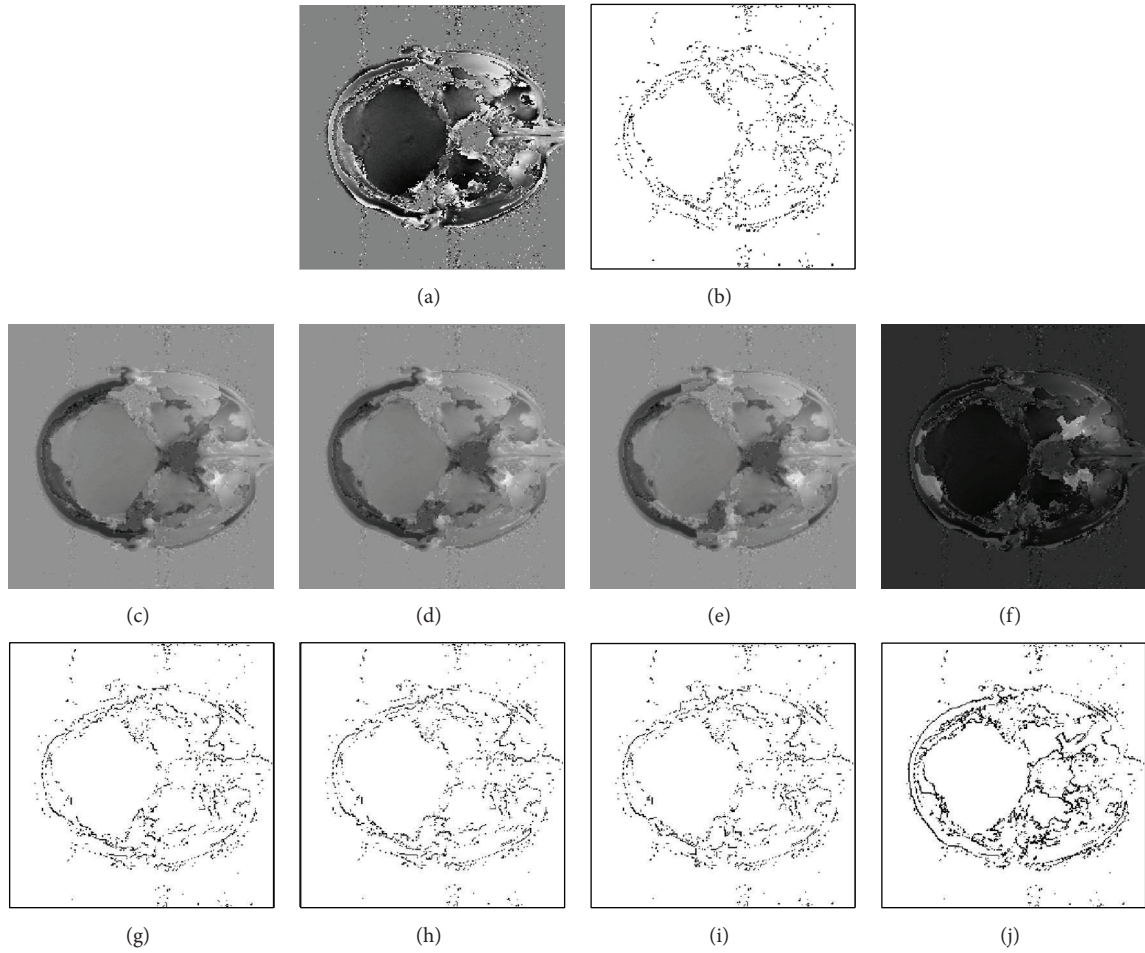


FIGURE 5: A  $256 \times 256$  MR head dataset, (a) its wrapped phases, and (b) its residue distribution including 1929 residues. Unwrapped phase images using the (c)  $SDML^P$ , (d) conventional minimum  $L^P$ -norm, (e) PUMA, and (f) PHUN methods. Discontinuities in the corresponding unwrapped phase images got by the (g)  $SDML^P$ , (h) conventional minimum  $L^P$ -norm, (i) PUMA, and (j) PHUN methods.

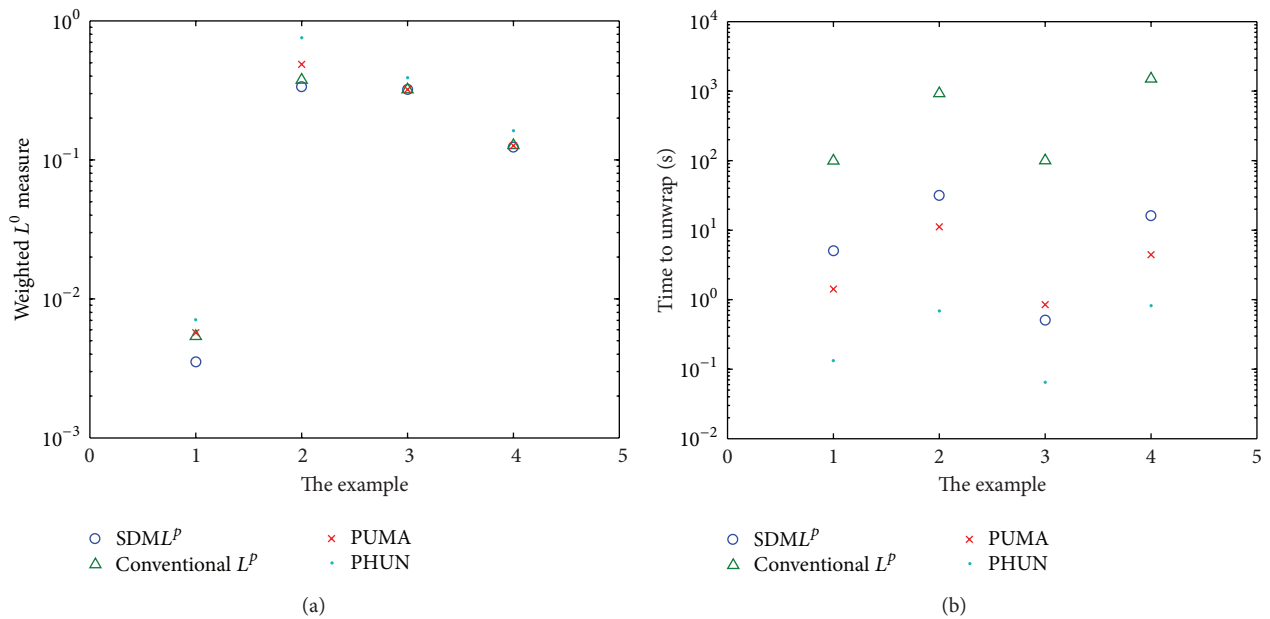


FIGURE 6: A comparison of the (a) weighted  $L^0$  measures and (b) execute time of the  $SDML^P$ , conventional minimum  $L^P$ -norm, PUMA, and PHUN methods in unwrapping the preceding four datasets.



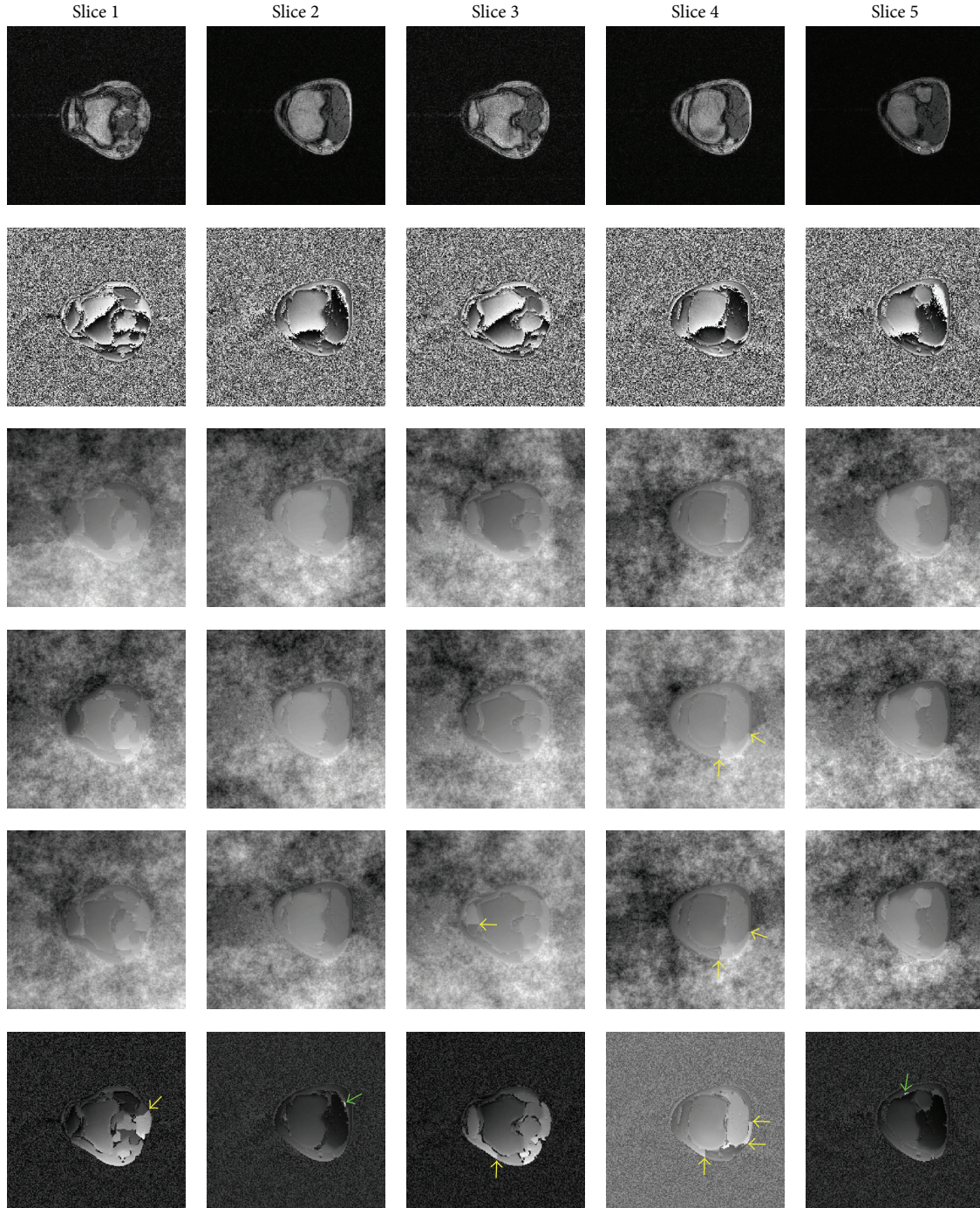


FIGURE 7: The transverse sections of a 5-slice MR knee dataset. The magnitude and wrapped phase images are shown in the first and second rows, respectively. Unwrapping results of the  $SDML^P$ , conventional minimum  $L^p$ -norm, PUMA, and PHUN methods are followed in a top-down order. Undesirable shear lines are marked with yellow arrows. The patches of outliers are marked with green arrows.

#### 4. Conclusions

In this work, we developed a sparse-representation-based direct minimum  $L^p$ -norm ( $SDML^P$ ) algorithm for the 2D phase unwrapping.

The user-defined weights are introduced in the objective function to improve the discontinuity-preserving ability of the  $SDML^P$  algorithm. Furthermore, the sparse structures

are used to represent the matrices involved in the objective function to accelerate the computation and decrease the memory space. Finally, the  $SDML^P$  algorithms computed effectively and efficiently by employing direct solvers.

The proposed algorithm does produce excellent, reliable results with a very small weighted  $L^0$  measure; it even allows phase images with large discontinuities through the whole phases to be unwrapped correctly. Moreover, benefiting from

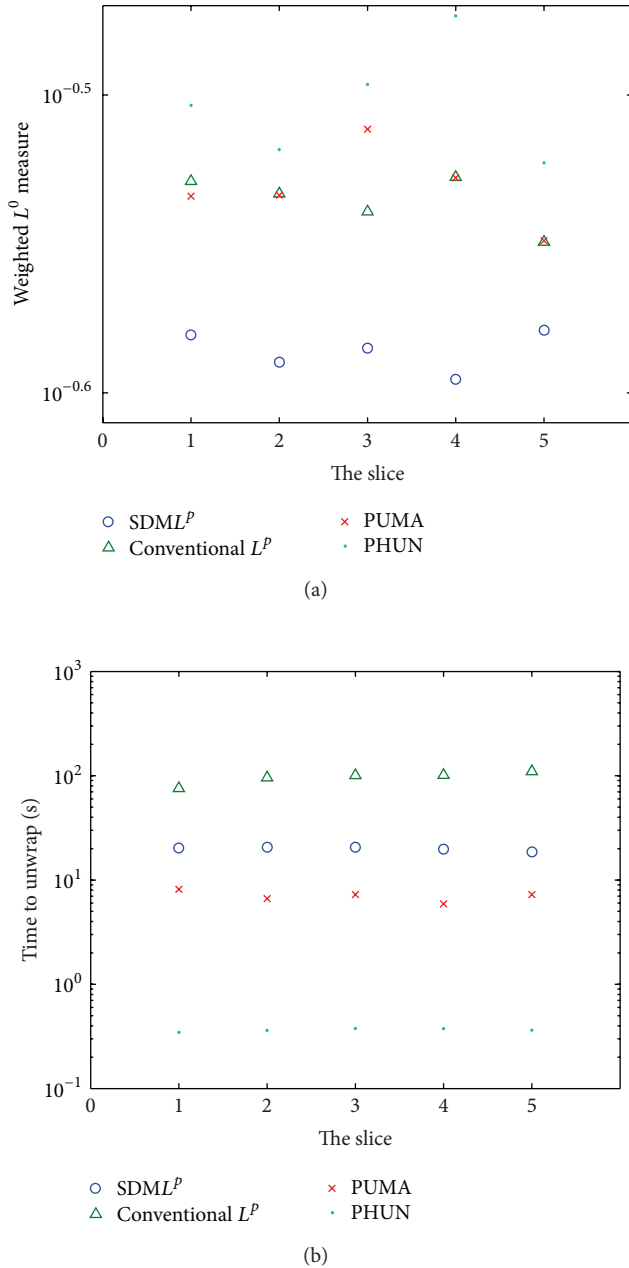


FIGURE 8: A comparison of the (a) weighted  $L^0$  measures and (b) execute time of the SDML<sup>P</sup>, conventional minimum  $L^P$ -norm, PUMA, and PHUN methods in unwrapping the multislice dataset.

using the sparse representation and well-developed direct solvers, the SDML<sup>P</sup> method converges much faster than the conventional minimum  $L^P$ -norm method. However, it is not fast enough. Further research can be devoted to reducing the execution time.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work was supported in part by the 973 National Basic Research & Development Program of China (2010CB732502) and the National Science & Technology Pillar Program (2011BAI12B01).

### References

- [1] J. Szumowski, W. R. Coshov, F. Li, and S. F. Quinn, "Phase unwrapping in the three-point Dixon method for fat suppression MR imaging," *Radiology*, vol. 192, no. 2, pp. 555–561, 1994.
- [2] A. Rauscher, M. Barth, J. R. Reichenbach, R. Stollberger, and E. Moser, "Automated unwrapping of MR phase images applied to BOLD MR-venography at 3 tesla," *Journal of Magnetic Resonance Imaging*, vol. 18, no. 2, pp. 175–180, 2003.
- [3] H. Bagher-Ebadian, Q. Jiang, and J. R. Ewing, "A modified fourier-based phase unwrapping algorithm with an application to MRI venography," *Journal of Magnetic Resonance Imaging*, vol. 27, no. 3, pp. 649–652, 2008.
- [4] P. Chiang, Y. Cai, K. H. Mak, and J. Zheng, "A B-spline approach to phase unwrapping in tagged cardiac MRI for motion tracking," *Magnetic Resonance in Medicine*, vol. 69, pp. 1297–1309, 2013.
- [5] P. Munger, G. R. Greller, T. M. Peters, and G. B. Pike, "An inverse problem approach to the correction of distortion in EPI images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 681–689, 2000.
- [6] X. Su and W. Chen, "Reliability-guided phase unwrapping algorithm: a review," *Optics and Lasers in Engineering*, vol. 42, no. 3, pp. 245–261, 2004.
- [7] M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," *Applied Optics*, vol. 41, no. 35, pp. 7437–7444, 2002.
- [8] S. Fang, L. Meng, L. Wang, P. Yang, and M. Komori, "Quality-guided phase unwrapping algorithm based on reliability evaluation," *Applied Optics*, vol. 50, no. 28, pp. 5446–5452, 2011.
- [9] K. Zhou, M. Zaitsev, and S. Bao, "Reliable two-dimensional phase unwrapping method using region growing and local linear estimation," *Magnetic Resonance in Medicine*, vol. 62, no. 4, pp. 1085–1090, 2009.
- [10] S. Witoszynskij, A. Rauscher, J. R. Reichenbach, and M. Barth, "Phase unwrapping of MR images using  $\Phi$ UN—a fast and robust region growing algorithm," *Medical Image Analysis*, vol. 13, no. 2, pp. 257–268, 2009.
- [11] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," *Radio Science*, vol. 23, no. 4, pp. 713–720, 1988.
- [12] D. C. Ghiglia and L. A. Romero, "Minimum  $L_p$ -norm two-dimensional phase unwrapping," *Journal of the Optical Society of America A*, vol. 13, no. 10, pp. 1999–2013, 1996.
- [13] T. Mario Costantini, "A novel phase unwrapping method based on network programming," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 3, pp. 813–821, 1998.
- [14] C. W. Chen and H. A. Zebker, "Network approaches to two-dimensional phase unwrapping: Intractability and two new algorithms," *Journal of the Optical Society of America A*, vol. 17, no. 3, pp. 401–414, 2000.
- [15] J. M. B. Dias and J. M. N. Leitão, "The  $\mathbb{Z}\pi$ M algorithm: a method for interferometric image reconstruction in SAR/SAS," *IEEE*



- Transactions on Image Processing*, vol. 11, no. 4, pp. 408–422, 2002.
- [16] J. M. Bioucas-Dias and G. Valadão, “Phase unwrapping via graph cuts,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 698–709, 2007.
- [17] K. Zhang, L. Ge, Z. Hu, A. H.-M. Ng, X. Li, and C. Rizos, “Phase unwrapping for very large interferometric data sets,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 4048–4061, 2011.
- [18] D. C. Ghiglia and L. A. Romero, “Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods,” *Journal of the Optical Society of America A*, vol. 11, no. 1, pp. 107–117, 1994.
- [19] G. Nico, G. Palubinskas, and M. Datcu, “Bayesian approaches to phase unwrapping: theoretical study,” *IEEE Transactions on Signal Processing*, vol. 48, no. 9, pp. 2545–2556, 2000.
- [20] L. Ying, Z.-P. Liang, D. C. Munson Jr., R. Koetter, and B. J. Frey, “Unwrapping of MR phase images using a Markov random field model,” *IEEE Transactions on Medical Imaging*, vol. 25, no. 1, pp. 128–136, 2006.
- [21] Z.-P. Liang, “A model-based method for phase unwrapping,” *IEEE Transactions on Medical Imaging*, vol. 15, no. 6, pp. 893–897, 1996.
- [22] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, John Wiley & Sons, New York, NY, USA, 1998.
- [23] T. J. Flynn, “Two-dimensional phase unwrapping with minimum weighted discontinuity,” *Journal of the Optical Society of America A*, vol. 14, no. 10, pp. 2692–2701, 1997.
- [24] G. Dardyk and I. Yavneh, “A multigrid approach to two-dimensional phase unwrapping,” *Numerical Linear Algebra with Applications*, vol. 11, no. 2-3, pp. 241–259, 2004.
- [25] I. Shalem and I. Yavneh, “A multilevel graph algorithm for two dimensional phase unwrapping,” *Computing and Visualization in Science*, vol. 11, no. 2, pp. 89–100, 2008.
- [26] P. Mistry, S. Braganza, D. Kaeli, and M. Leeser, “Accelerating phase unwrapping and affine transformations for optical quadrature microscopy using CUDA,” in *Proceedings of the 2nd Workshop on General Purpose Processing on Graphics Processing Units (GPGPU '09)*, pp. 28–37, Washington, DC, USA, March 2009.
- [27] R. P. Tewarson, *Sparse Matrices*, Academic Press, New York, NY, USA, 1973.
- [28] U. W. Pooch and A. Nieder, “A survey of indexing techniques for sparse matrices,” *ACM Computing Surveys*, vol. 5, pp. 109–133, 1973.
- [29] J. A. Scott and Y. Hu, “Experiences of sparse direct symmetric solvers,” *ACM Transactions on Mathematical Software*, vol. 33, no. 3, Article ID 1268772, p. 18, 2007.
- [30] D. S. Watkins, *Fundamentals of Matrix Computations*, vol. 64, John Wiley & Sons, New York, NY, USA, 2nd edition, 2002.
- [31] A. Krishnamoorthy and D. Menon, “Matrix inversion using Cholesky decomposition,” <http://arxiv.org/abs/1111.4144>.
- [32] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, New York, NY, USA, 1986.
- [33] N. I. M. Gould, J. A. Scott, and Y. Hu, “A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations,” *ACM Transactions on Mathematical Software*, vol. 33, no. 2, Article ID 1236465, p. 10, 2007.
- [34] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, “Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate,” *ACM Transactions on Mathematical Software*, vol. 35, no. 3, article 22, pp. 1–14, 2008.
- [35] I. S. Duff, “MA57—a code for the solution of sparse symmetric definite and indefinite systems,” *ACM Transactions on Mathematical Software*, vol. 30, no. 2, pp. 118–144, 2004.
- [36] S. Chavez, Q.-S. Xiang, and L. An, “Understanding phase maps in MRI: a new outline phase unwrapping method,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 966–977, 2002.
- [37] K. Chen, J. Xi, Y. Yu, and J. F. Chicharo, “Fast quality-guided flood-fill phase unwrapping algorithm for three-dimensional fringe pattern profilometry,” in *Optical Metrology and Inspection for Industrial Applications*, vol. 78550X of *Proceedings of the SPIE*, Beijing, China, October 2010.
- [38] W. He, Y. Cheng, L. Xia, and F. Liu, “A new particle swarm optimization-based method for phase unwrapping of MRI data,” *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 475745, 9 pages, 2012.
- [39] B. Spottiswoode, “2D phase unwrapping algorithms,” <http://www.mathworks.com/matlabcentral/fileexchange/22504>.