Medical Image Analysis 52 (2019) 13–23

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Quality-based *UnwRap of SUbdivided Large Arrays* (URSULA) for high-resolution MRI data ☆

J. Lindemeyer [a,*], A.-M. Oros-Peusquens [a], N.J. Shah [a,b,c,d,e]

[a] Institute of Neuroscience and Medicine – 4, Forschungszentrum Jülich GmbH, Jülich 52425, Germany
[b] Department of Neurology, Faculty of Medicine, RWTH Aachen University, Aachen 52074, Germany
[c] JARA – BRAIN – Translational Medicine, RWTH Aachen University, Aachen 52074, Germany
[d] Institute of Neuroscience and Medicine – 11, Forschungszentrum Jülich GmbH, Jülich 52425, Germany
[e] Monash Biomedical Imaging, School of Psychological Sciences, Monash University, Melbourne, Victoria, Australia

ARTICLE INFO

Article history:
Received 21 November 2017
Revised 17 October 2018
Accepted 12 November 2018
Available online 13 November 2018

Keywords:
MRI
Phase unwrapping
Three dimensional unwrapping
Parallel computing

ABSTRACT

In Magnetic Resonance Imaging, mapping of the static magnetic field and the magnetic susceptibility is based on multidimensional phase measurements. Phase data are ambiguous and have to be unwrapped to their true range in order to exhibit a correct representation of underlying features. High-resolution imaging at ultra-high fields, where susceptibility and phase contrast are natural tools, can generate large datasets, which tend to dramatically increase computing time demands for spatial unwrapping algorithms. This article describes a novel method, URSULA, which introduces an artificial volume compartmentalisation that allows large-scale unwrapping problems to be broken down, making URSULA ideally suited for computational parallelisation. In the presented study, URSULA is illustrated with a quality-guided unwrapping approach. Validation is performed on numerical data and an application on a high-resolution measurement, at the clinical field strength of 3T is demonstrated. In conclusion, URSULA allows for a reduction of the problem size, a substantial speed-up and for handling large data sets without sacrificing the overall accuracy of the resulting phase information.

1. Introduction

Multidimensional phase data acquired in Magnetic Resonance Imaging (MRI) reflect the position-dependent, in-plane orientation of the transverse magnetisation, observed within a rotating frame of reference. In a hypothetical ideal experiment on a homogeneous medium, the phase is expected to be constant (zero) throughout the volume. Under realistic conditions, however, phase data reflect inhomogeneities caused by the transmit and receive processes, by the distribution of the static magnetic field and by local electromagnetic tissue properties, such as the magnetic susceptibility and chemical shifts. Furthermore, information encoded in the phase data can be used for complex applications such as flow imaging (e.g. Lotz et al., 2002) or Quantitative Susceptibility Mapping (QSM: Wang and Liu, 2015; Reichenbach, 2012), the latter being derived from local changes in the static magnetic field.

The projection of the magnetisation state observed in MRI is restricted to the complex plane – implying a range of $[-\pi, \pi)$ (or $[0, 2\pi)$ respectively), such that the actual evolution of the magnetisation over time is masked, whenever this range is exceeded (e.g. Reichenbach et al., 1997). A globally consistent representation of the true phase, biased only by a potential offset of $n \cdot 2\pi$, can be restored by spatial or temporal unwrapping (e.g. Cusack and Papadakis, 2002 or Haacke et al., 1999, Chapter 25.4). Datasets used for QSM are typically acquired with gradient recalled echo (GRE) or related 3D sequences (Reichenbach, 2012). These data typically have four dimensions – three spatial and optionally, a temporal one. One-dimensional unwrapping along the temporal dimension is the most straightforward approach (e.g. Windischberger et al., 2004; Dagher et al., 2014). However, in order to achieve a high signal-to-noise ratio, experiments are performed with low bandwidth, implying wide echo spacing (e.g. Deistung et al., 2008). This tends to cause undersampling with respect to the Nyquist sampling limitation in time domain in locations where the phase evolves rapidly. Thus, in many

Abbreviations: AR, Abdul-Rahman unwrapping; GRE, Gradient Recalled Echo; MRI, Magnetic Resonance Imaging; QBU, Quality Based Unwrapping; QSM, Quantitative Susceptibility Mapping; URSULA, UnwRap of SUbdivided Large Array.

☆ **Conflicts of interest:** No author associated with this paper has disclosed any potential or pertinent conflicts of interest.

* Corresponding author.
E-mail addresses: j.lindemeyer@fz-juelich.de (J. Lindemeyer), a.m.oros-peusquens@fz-juelich.de (A.-M. Oros-Peusquens), n.j.shah@fz-juelich.de (N.J. Shah).

https://doi.org/10.1016/j.media.2018.11.004
1361-8415/

cases, unwrapping in the spatial domain is the method of choice, posing a three-dimensional unwrapping problem (e.g. Jenkinson, 2003).

A variety of approaches have been discussed in the literature of the last decades for two (Goldstein et al., 1988; Hedley and Rosenfeld, 1992; Strand et al., 1999; Strand and Taxt, 2002; Karout et al., 2007) or three (Witoszynskyj et al., 2009; Dong and Ji, 2010; Feng et al., 2012; Liu et al., 2012; Li et al., 2014) dimensions. One of the most popular algorithms, PRELUDE (Jenkinson, 2003), uses a compartmentalising strategy, dividing the phase volume into phase segments of a certain value range and matching these regions. More consistent and time-efficient unwrapping results can be achieved by quality-based unwrapping (QBU) algorithms such as those presented by Cusack and Papadakis (2002) or Abdul-Rahman et al. (2007). The latter method finds strong resonance in the QSM community (e.g. Schweser et al., 2011; Liu et al., 2011; Haacke et al., 2015) and serves as the reference method herein.

With the aim of processing large matrices acquired for high-resolution data, e.g. for ultra-high field MRI systems, the time efficiency of unwrapping becomes a critical point. Fast unwrapping is essential for applications with compulsory consecutive processing steps, such as QSM. Particularly, in order to establish phase-based methods in clinical practice, where physicians require immediate results to decide upon further measurements or surgical options, short computing times are required.

In the following, a novel technique, UnwRap of SUbdivided Large Arrays (URSULA), is presented (first mentioned in Lindemeyer et al., 2013 and Lindemeyer, 2015). It combines the high reliability of the QBU algorithm proposed by Abdul-Rahman et al. (2007) with a parallelisation technique that allows the distribution of the unwrapping task to a parallel computing grid. This is an important development, given the fact that the original algorithm is not well suited for parallel computing in its basic form. In contrast, URSULA can reduce the computing time significantly, especially for extensive volumes, while still preserving the spatial consistency of the result. Beyond the QBU algorithm used in this work, URSULA offers a general means of parallelisation for any existing unwrapping approach applied to any given scenario.

2. Materials and methods

2.1. Theory:

Abdul-Rahman et al. (2007) describe an approach to spatially unwrap phase data based on QBU. A quality describing the phase change along all directions is assigned to each voxel by calculating *second differences* within a ± 1 neighbourhood. Subsequently, the quality of a surface between two adjacent voxels is calculated by adding the quality of both. To solve the global unwrapping problem, all surfaces – or *links* – are processed in ascending order, or equivalently in descending quality. Voxels connected by the currently processed link are grouped. In this way, voxels and groups join other groups until only one global volume remains. Sequential computing of the interfaces is implied by this strategy. The Abdul–Rahman approach will henceforth be denoted by AR.

The concept behind the present algorithm is to distribute the unwrapping problem by subdividing the entire volume into a number of subsets. Every subset is processed as an individual unwrap volume. This decreases the size of each individual unwrapping problem dramatically and allows for parallel computation of the independent problems. The results are reassembled and matched, as illustrated in Fig. 1. Since the actual unwrapping problem favours sequential solving, a careful choice for subdivision and reassembling of the volume is essential for a consistent result. One could choose a subdivision that reduces the dimensionality by processing two-dimensional slices of the volume independently, as in classical applications. Yet, when excluding one dimension, the information contained is lost and cannot contribute to a proper solution. This becomes fatal whenever rapid phase changes within the slice prevent proper unwrapping, whilst a moderate phase gradient in slicing direction would still suffice to solve the problem. Furthermore, evaluation regions of non-convex shape tend to become fragmented into disjoint segments when split into slices.

In the most popular region-merging approach, PRELUDE (Jenkinson, 2003), datasets are compartmentalised according to the phase behaviour, and hence the merging process is the demanding step for computation. Parallelising this step is difficult, as it requires permanent inter-process communication with all subsets. Our approach aims to relocate expensive computing to the intra-subset unwrapping part so that this step can be parallelised. A low-cost matching process is achieved by an optimised subset shaping. Thus, the present approach uses three-dimensional subsets aligned on a regular, orthogonal grid. In the literature one can find some related concepts, e.g. Strand et al. (1999) and Strand and Taxt (2002), where rectangular subdivision is used for phase unwrapping in two dimensions, however a different strategy is employed.

Let $\varphi[ijk]$ be a discretised phase volume of size $N_x \times N_y \times N_z$, and m be the volume mask. The subdivision will be defined by the subset size $s_x \times s_y \times s_z$, leading to the following subset count:

$$c_x = \lceil N_x/s_x \rceil, \quad c_y = \lceil N_y/s_y \rceil, \quad c_z = \lceil N_z/s_z \rceil \tag{1}$$

(where $\lceil \cdot \rceil$ implies rounding to the upper integer). The remaining voxels from non-integer division results are assigned to an additional subset with appropriately reduced dimensioning. A voxel in the original volume is addressed as $\varphi[ijk]$, while the same voxel expressed within the subset uvw, located at position $[pqr]$, is denoted by:

$$\varphi_{uvw}\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \varphi\begin{bmatrix} u \cdot s_x + p \\ v \cdot s_y + q \\ w \cdot s_z + r \end{bmatrix} = \varphi\begin{bmatrix} i \\ j \\ k \end{bmatrix}, \tag{2}$$

with $p \in [0{:}\ s_x)$, $q \in [0{:}\ s_y)$, $r \in [0{:}\ s_z)$ and $u \in [0{:}\ c_x)$, $v \in [0{:}\ c_y)$, $w \in [0{:}\ c_z)$. This indexing scheme is illustrated in Fig. 2. Subset masks $m_{uvw}[pqr]$ are defined equivalently, based on the volume mask m.

Each subset, φ_{uvw}, is processed with AR, resulting in the subset unwrap results $\tilde{\varphi}_{uvw}$:

$$\tilde{\varphi}_{uvw} = \mathrm{AR}(\varphi_{uvw}) \tag{3}$$

For this step, the subset definition is temporarily extended by one voxel in all directions ($p \in [-1 : s_x + 1)$, $q \in [-1 : s_y + 1)$ and $r \in [-1 : s_z + 1)$). This ensures accurate results on the subset surfaces, as all voxels of the actual subset possess a complete neighbourhood. An unwrapped subset $\tilde{\varphi}_{uvw}$ may exhibit a mismatch of constant offset $\pm[1, 2, 3, \ldots] \cdot 2\pi$ with respect to the global unwrap, and thus to other subsets, resulting from the natural independence of the applied unwrapping processes (visible in Fig. 1):

$$m_{uvw} \cdot \tilde{\varphi}_{uvw,\,\mathrm{global}} = m_{uvw} \cdot \tilde{\varphi}_{uvw} + 2\pi \cdot n_{uvw}, \tag{4}$$

where n_{uvw} represents the *wrap count*. It is necessary to determine the wrap count for all subsets and this process is known as *matching*.

When using a brain mask, sequential matching of neighbouring subsets cannot generally solve the global unwrapping problem. Fig. 3a-b illustrates subsets aligned in one dimension, but the missing mask support on their direct connection cannot be matched unless the connection is established within the mask support.

Hence, a quality-based, surface-matching algorithm was developed. The problem is related to unwrapping on voxel-scale, as the subsets differ by multiples of 2π. This motivates the employment of the AR solving principle, transferred to a macroscopic view

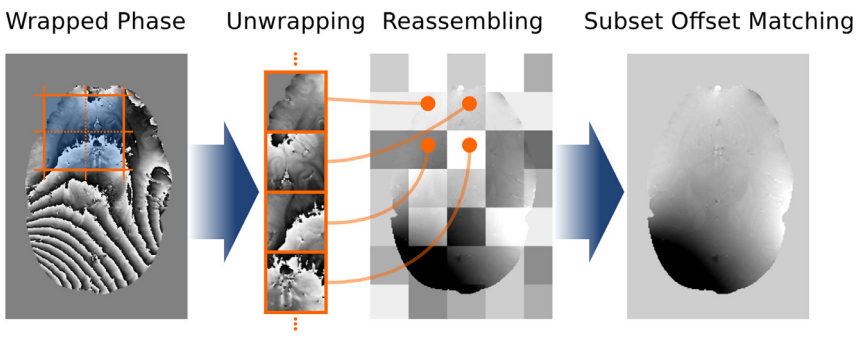

Fig. 1. Schematic sketch of URSULA workflow. This illustration is adapted from Lindemeyer (2015), p. 76, Fig. 5.1.

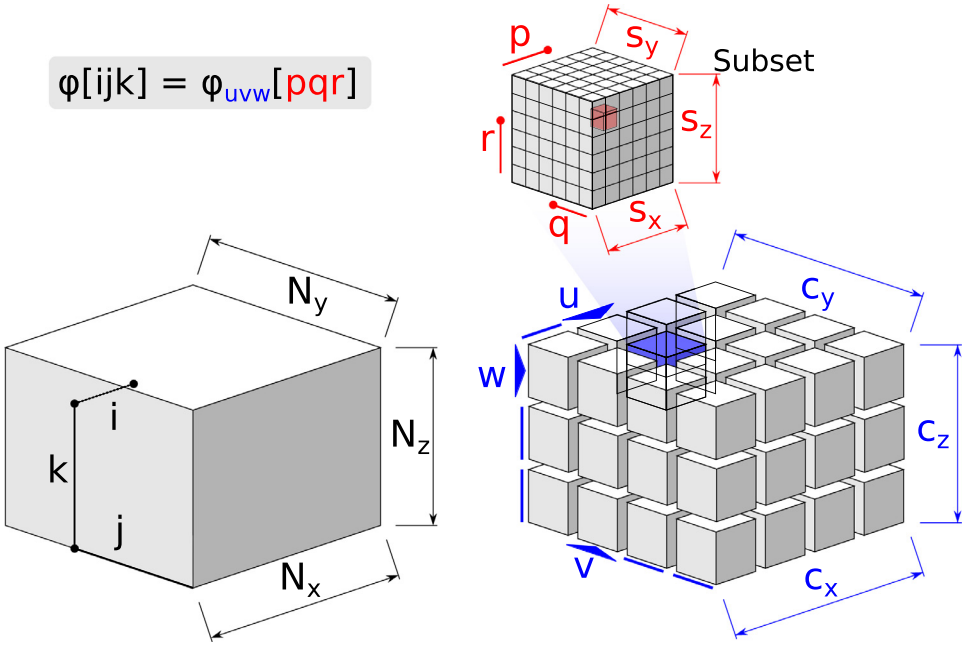

Fig. 2. Indexing scheme for global (left),subset (right) and intra-subset (top) reference.

where the subsets act as base elements and links are defined by their shared surface. For example, two adjacent subsets in the x-direction share the following valid voxel neighbours:

$$\tilde{\varphi}_{u,v,w}\begin{bmatrix} s_x-1 \\ q \\ r \end{bmatrix} \overset{\text{neighb.}}{\longleftrightarrow} \tilde{\varphi}_{u+1,v,w}\begin{bmatrix} 0 \\ q \\ r \end{bmatrix}$$

$$\text{if}: \; m_{u,v,w}\begin{bmatrix} s_x-1 \\ q \\ r \end{bmatrix} \cdot m_{u+1,v,w}\begin{bmatrix} 0 \\ q \\ r \end{bmatrix} = 1 \tag{5}$$

In order to maximise the number of reference voxel pairs, surfaces fully covered by the mask are strongly favoured over partially covered ones. Thus, the quality measure employed for the matching order should be the mask coverage of the link surface. For each subset a so-called connectivity (percentage of the subset edges shared with neighbouring subsets on the mask support) is estimated by conjunction of the proximate subset surface masks. For example, in the first index direction (x) this means:

$$Q_x(u,v,w) = \sum_{q,r}\left(m_{u,v,w}\begin{bmatrix} s_x-1 \\ q \\ r \end{bmatrix} \cdot m_{u,v+1,w}\begin{bmatrix} 0 \\ q \\ r \end{bmatrix}\right) \tag{6}$$

The wrap distance $\Delta n_x(u, v, w)$ between the subsets is calculated accordingly:

$$\Delta n_x(u,v,w) = \text{round}\left(\sum_{q,r}\left[\left(m_{uv,w,w}\begin{bmatrix} s_x-1 \\ q \\ r \end{bmatrix} \cdot m_{u+1,v,w}\begin{bmatrix} 0 \\ q \\ r \end{bmatrix}\right) \cdot \left(\tilde{\varphi}_{u,v,w}\begin{bmatrix} s_x-1 \\ q \\ r \end{bmatrix} - \tilde{\varphi}_{u+1,v,w}\begin{bmatrix} 0 \\ q \\ r \end{bmatrix}\right)\right] \cdot (2\pi \cdot Q_x(u,v,w))^{-1}\right) \tag{7}$$

Qualities and wrap distances for y and z direction can be derived respectively.

Fig. 3c illustrates the quality estimation of subset connectivities. The matching procedure follows the iterative grouping approach of AR. Links are processed with decreasing surface coverage, Q_x, Q_y or Q_z. Every two adjacent subsets **a** and **b** are processed as follows:

- *Neither* ***a*** *nor* ***b*** *is assigned to a group:*
  Assign new group identifier, g, and shift one subset by the mutual phase difference Δn.
- *Subset* ***a*** *belongs to group* g_a, **b**, *is unassigned:*
  Subset **b** joins group g_a ($g_a = g_a \cup \mathbf{b}$), while being shifted by Δn.
- *Both grouped,* $\mathbf{a} \in g_a$ and $\mathbf{b} \in g_b$:
  Smaller group, e.g. $\#g_a \leq \#g_b$, joins the bigger one, $\mathbf{g_b} = \mathbf{g_b} \cup \mathbf{g_a}$, with every assigned subset in g_a being shifted by Δn.
- *Both belong to the same group* ($g_a = g_b$): No action.

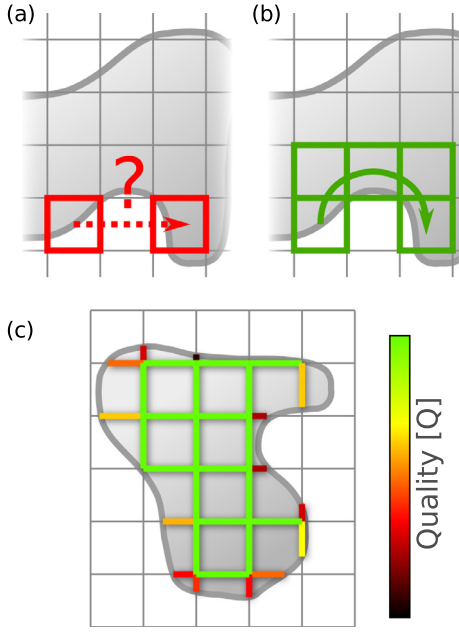

Fig. 3. Subset matching: (a) sequential matching fails due to a lack of mask support (grey); (b) following the mask support – allows for valid results; (c) scheme of assigning edge qualities based on mask coverage. In the matching process, surfaces of high quality are processed first. This illustration is partially adapted from Lindemeyer (2015), p. 78, Fig. 5.3.

The relative wrap distances, Δn_{uvw}, are collected in a global wrap distance map, n_{uvw}. Once all subset links have been processed, n_{uvw} is applied to the volume, resulting in a consistent unwrap:

$$m_{uvw}\begin{bmatrix}p\\q\\r\end{bmatrix}\cdot\tilde{\varphi}_{uvw}\begin{bmatrix}p\\q\\r\end{bmatrix}+2\pi\cdot n_{uvw}\longrightarrow\tilde{\varphi}\begin{bmatrix}i\\j\\k\end{bmatrix} \tag{8}$$

Fig. 4 illustrates the workflow of the URSULA matching algorithm. Potential remaining global offsets due to unwrapping and matching are avoided by calculating a local unwrap of a central region of interest, $\mathcal{R}$, prior to and after the URSULA parallelisation and matching. The unwrapped volume is shifted by $2\pi\cdot n_{\text{global}}$, where:

$$n_{\text{global}}=\text{round}\left(\frac{\overline{\text{AR}(\varphi_{\text{post}}[\mathcal{R}])-\text{AR}(\varphi_{\text{prior}}[\mathcal{R}])}}{2\pi}\right) \tag{9}$$

Hence, the global phase unwrap is:

$$\varphi_{\text{global}}=\tilde{\varphi}-2\pi\cdot n_{\text{global}} \tag{10}$$

2.2. Efficiency

The AR is expected to scale roughly with $t_{\text{AR}}=\alpha_{\text{AR}}\cdot N_xN_yN_z\approx\alpha_{\text{AR}}\cdot N^3$, where α_{AR} is a proportionality coefficient. If k nodes are available to compute $l=c_xc_yc_z$ subsets, the ideal time for unwrapping within URSULA would be $t_{\text{unw}}=\alpha_{\text{AR}}\cdot(l/k)\cdot s_xs_ys_z$ without considering matching and overhead. The time required by the matching process is proportional to the number of subsets and to their surface: $t_{\text{match}}\approx\alpha_{\text{match}}\cdot l\cdot(s_xs_ys_z)^{2/3}$. Overhead time for distribution and collection of parallelised computing results scales with subset size and the number of computing nodes, $t_{\text{oh}}=\alpha_{\text{oh}}\cdot l\cdot s_xs_ys_z$. The expected behaviour for the URSULA computing time thus calculates to:

$$t_{\text{URS}}=\alpha_{\text{AR}}\cdot\frac{l}{k}\cdot s_xs_ys_z+\alpha_{\text{oh}}\cdot l\cdot s_xs_ys_z+\alpha_{\text{match}}\cdot l\cdot(s_xs_ys_z)^{2/3}$$

$$\approx\frac{t_{AR}}{k}+\alpha_{\text{oh}}\cdot N^3+\alpha_{\text{match}}\cdot l\cdot s^2 \tag{11}$$

Hence, AR and URSULA both predominantly scale with N^3 (N being the voxel count in one dimension). The transfer overhead, scaling also with N^3, counteracts the time reduction due to parallelisation. Including a constant offset, γ, to account for any overhead induced by pre-compiling and executing the comprehensive function leaves $\alpha\cdot N^3+\beta\cdot ls^2+\gamma$ for URSULA, with α,β,γ being fitting parameters.

2.3. Implementation

The actual AR algorithm is available as a C-based Python wrapper package, thus an implementation of URSULA in Python was developed to facilitate a fair comparison and to allow for easy implementation in other projects. The URSULA package uses the AR algorithm as a back-end for unwrapping the individual subsets. Listing 1 shows URSULA in simplified pseudo-code. In Listing 2 the subset matching process is described. For distributed computing, the package *dispy* (dispy.sourceforge.net) was chosen, as it can easily be set up on any number of computers connected within a local area network.

Algorithm 1 URSULA (simplified).

procedure URSULA($\varphi, m, s_x, s_y, s_z$)
  $N_{x,y,z} \leftarrow \text{shape}(\varphi)$
  $c_{x,y,z} \leftarrow \text{ceil}(N_{x,y,z}/s_{x,y,z})$
  $a_{x,y,z}[pqr] \leftarrow$ 3D grid $(0 : s_{x,y,z} : N_{x,y,z})$
  $b_{x,y,z}[pqr] \leftarrow a_{x,y,z}[pqr] + s_{x,y,z}$
  if $b_{x,y,z}[pqr] > N_{x,y,z}$ **then**
    $b_{x,y,z}[pqr] \leftarrow N_{x,y,z}$
  $\mathcal{R} \leftarrow$ central ROI
  $\tilde{\varphi}_{\text{pre}} \leftarrow \text{AR}(\varphi[\mathcal{R}])$
  $\tilde{\varphi} \leftarrow$ 0-array of size φ
  ▷ Process in parallel:
  for $i_x, i_y, i_z \in [1 : c_x, 1 : c_y, 1 : c_z]$ **do**
    $\varphi_{\text{sub}} \leftarrow \varphi[a_{x,y,z}[i_xi_yi_z] - 1 : b_{x,y,z}[i_xi_yi_z] + 1]$
    $\tilde{\varphi}_{\text{sub}} \leftarrow \text{AR}(\varphi_{\text{sub}})$
    $\tilde{\varphi}[a_{x,y,z}[i_xi_yi_z] : b_{x,y,z}[i_xi_yi_z]]$
      $\leftarrow \tilde{\varphi}_{\text{sub}}[1 : s_{x,y,z} + 1]$
  $\tilde{\varphi} \leftarrow \text{subsetmatch}(\tilde{\varphi}, m, s_x, s_y, s_z)$
  $\tilde{\varphi}_{\text{post}} \leftarrow \text{AR}(\tilde{\varphi}[\mathcal{R}])$
  $n_{\text{global}} \leftarrow \text{round}((\overline{\tilde{\varphi}_{\text{post}} - \tilde{\varphi}_{\text{pre}}})/2\pi)$
  $\tilde{\varphi} \leftarrow \tilde{\varphi} - n_{\text{global}} \cdot 2\pi$
return $\tilde{\varphi}$

2.4. Hardware

Processing took place on up to six nodes of a Linux-based computing cluster consisting of quad-core processors at 2.4 GHz with 8 GB of memory and eight virtual cores. The nodes communicate via Gigabit Ethernet connection.

2.5. Simulation

For the purpose of validation, a numerical phantom is employed. Since the number of computed interfaces is independent of the phase data contained, the computing time required by AR and URSULA depends predominantly on the volume size. Thus, without loss of generality, one particular function is chosen, challenging the algorithm with high wrap frequency, but controlled phase changes within the simulated mask, keeping the spatial distance of phase wraps in Nyquist-compliant range for at least one dimension everywhere:

$$f = x - 2y + z + 0.01\cdot x^2 - 0.01\cdot(z^2 - y^2)$$

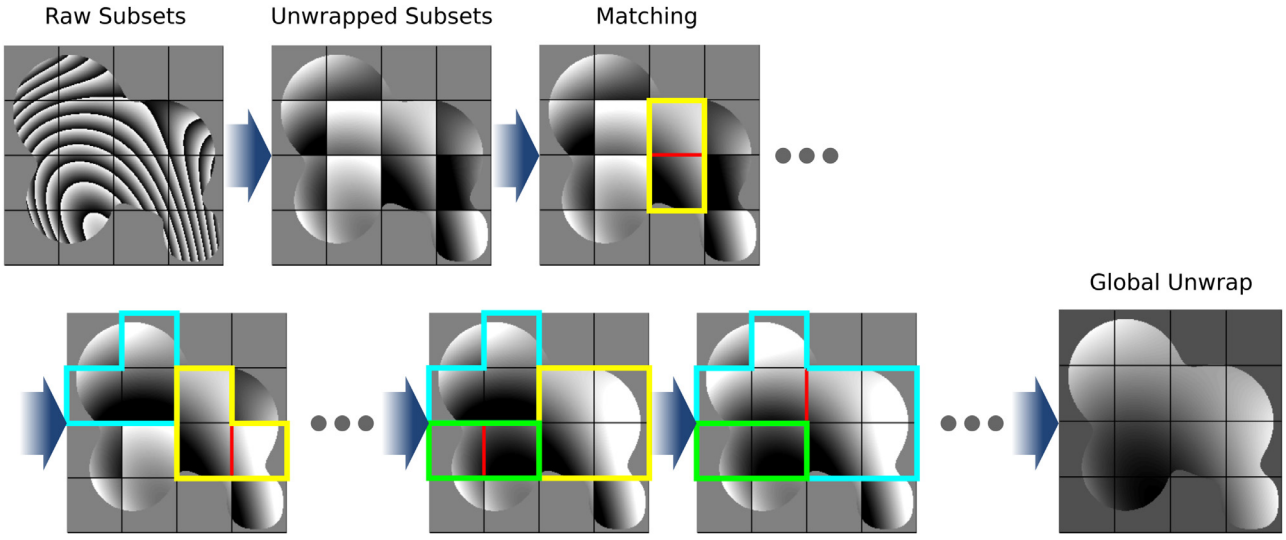

Fig. 4. Schematic example for matching and grouping. From left to right, top to bottom: Raw phase (subdivided), unwrapped subsets, matching and grouping steps and rightmost the unwrap result after matching all subsets. The red line indicates the current interface being computed, while the coloured borders mark different groups. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

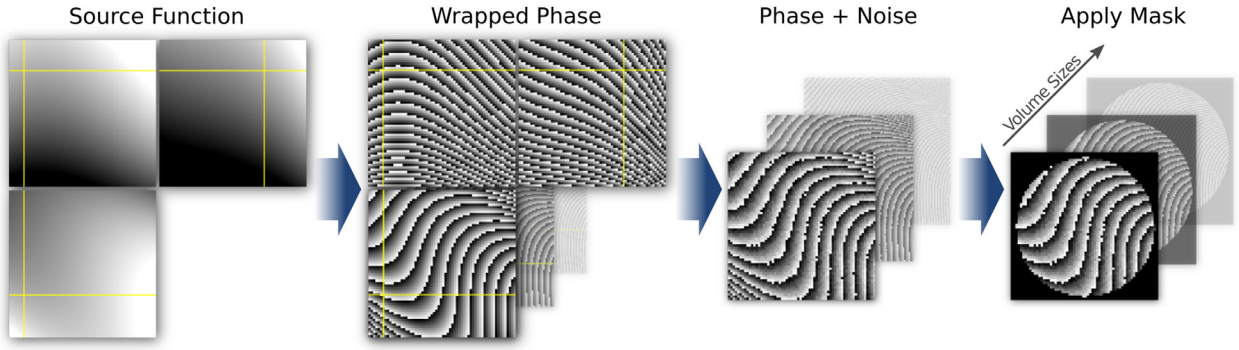

Fig. 5. Simulation workflow showing sample slices at a volume size of 64^3 (128^3 and 256^3 in the background, respectively) for all simulation steps.

$$+0.0004 \cdot (z-x)^3 - 0.0003 \cdot y^3, \tag{12}$$

with $x, y, z \in [-1, 1]$. For a grid of size $N \times N \times N$ the canonical variables are scaled by the factor γ/N, where $\gamma = 64$.

The function is chosen to behave smoothly within the evaluation mask, m, defined further below. Phase noise, $\delta\varphi$, is generated with normal distribution at a standard deviation of $\sigma_\varphi = 0.25$ rad. The simulated and the wrapped phase are obtained by:

$$\varphi_{\text{true}} = 0.6 \cdot \frac{N}{\gamma} \cdot f + \delta\varphi \tag{13}$$

$$\varphi = \text{mod}_{2\pi}(\varphi_{\text{true}}), \tag{14}$$

where 0.6 is a scaling factor and multiplication with N/γ ensures to approximately scale the spatial wrap distance (phase change per voxel) with the volume dimensioning.

A mask is defined as:

$$m = \left(\sqrt{x^2 + y^2 + z^2} < \frac{N}{1.8}\right), \tag{15}$$

and set *False* at the volume surface ($x, y, z = 0$ or $N - 1$)

A comparison based on simulation between AR and URSULA is performed with $\sigma_\varphi = 0.25$ rad for $N = [64, 80, \ldots, 336, 352]$ – examples for the synthetic data are shown in Fig. 5.

Phase unwrapping of φ is performed using AR on a single node, and with URSULA on 1, 2, 4, and 6 nodes with each 4 dispynode processes running in parallel. For parameter analysis, URSULA is applied with subset sizes between $N/8$ and $N/2$ in steps of 8 voxels. For each parameterisation 30 averages are computed.

The computing time is recorded and the accuracy of the unwrapping results per voxel and cycle (2π) is calculated by:

$$e_\varphi = \frac{\sum_{(x,y,z)\,\in m} \left|\tilde{\varphi}_{\text{unw}} - \varphi_{\text{true}} - 2\pi \cdot n_{\text{d}}\right|}{2\pi \cdot \sum_{(x,y,z)\,\in m} 1}, \tag{16}$$

where n_{d} is the estimated global phase difference between result and ground truth in multiples of 2π.

Additionally, the noise influence is assessed using simulations at $\sigma_\varphi = [0.1, 0.2, \ldots, 1.0]$ rad at an array size of 256^3 and subset dimensioning of $[16^3, 24^3, \ldots, 64^3]$ with 30 averages each.

2.6. Sample application

In order to demonstrate the method in the context of typical data, a healthy volunteer was measured on a Siemens 3 T human scanner (Siemens Healthcare, Erlangen, Germany) after obtaining informed consent. A standard 3D multiple-echo gradient-echo sequence with TE = [9.68, 31.49] ms, TR = 42 ms, $\alpha = 14°$ and BW = 70 Hz/Px was acquired. The resolution was 0.6 mm isotropic and the matrix size $310 \times 320 \times 160$ voxel. Standard shimming routines of the manufacturer were employed. The second echo was chosen for evaluation. Furthermore, by artificially multiplying the phase values by a factor of two, a more challenging scenario was gen-

Algorithm 2 Subsetmatch (simplified).

procedure SUBSETMATCH($\tilde{\varphi}, m, s_x, s_y, s_z$)
  $N_{x,y,z} \leftarrow \text{shape}(m)$
  $c_{x,y,z} \leftarrow \text{ceil}(N_{x,y,z}/s_{x,y,z})$
  $a_{x,y,z}[pqr] \leftarrow$ 3D grid $(0 : s_{x,y,z} : N_{x,y,z})$
  $b_{x,y,z}[pqr] \leftarrow a_{x,y,z}[pqr] + s_{x,y,z}$
  if $b_{x,y,z}[pqr] > N_{x,y,z}$ **then**
    $b_{x,y,z}[pqr] \leftarrow N_{x,y,z}$
  $Q_x \leftarrow$ 0-array in size $[c_x - 1, c_y, c_z]$
  $Q_y \leftarrow$ 0-array in size $[c_x, c_y - 1, c_z]$
  $Q_z \leftarrow$ 0-array in size $[c_x, c_y, c_z - 1]$
  for $i_x \in [0 : c_x]$, $i_y \in [0 : c_z]$, $i_z \in [0 : c_z]$ **do**
    if $i_x < c_x$ **then**
      $Q_x[i_x i_y i_z] \leftarrow \text{surf}([i_x, i_y, i_z], [i_x + 1, i_y, i_z])$
    if $i_y < c_y$ **then**
      $Q_y[i_x i_y i_z] \leftarrow \text{surf}([i_x, i_y, i_z], [i_x, i_y + 1, i_z])$
    if $i_z < c_z$ **then**
      $Q_z[i_x i_y i_z] \leftarrow \text{surf}([i_x, i_y, i_z], [i_x, i_y, i_z + 1])$
  $F \leftarrow$ 0-array of size $[5, (c_x - 1)c_y c_z + c_x(c_y - 1)c_z + c_x c_y(c_z - 1)]$
  $F[0, :] \leftarrow [Q_x[:], Q_y[:], Q_z[:]]$
  $F[1 : 3, :] \leftarrow$ interface position
  $F[4, :] \leftarrow$ interface direction
  $F \leftarrow \text{sort}(F, \text{by column } 0, \text{inverse})$ ▷ Sort by Quality
  $g \leftarrow (-1)$-array in size $[c_x, c_y, c_z]$
  $n \leftarrow$ 0-array in size $[c_x, c_y, c_z]$
  for i = Tuples in F **do**
    if $i[0] \leq 0.0$ **then** skip to next
    $[k, k_{\text{rel}}] \leftarrow$ coordinate/direction $F[1 : 4, i]$
    $\varphi_{\text{off}} \leftarrow$ difference to neighbour (Eq. 7)
    $n_{\text{tmp}} = \leftarrow n[k + k_{\text{rel}}] - n[k] + \text{round}(\varphi_{\text{off}}/2\pi)$
    $v_1, v_2 \leftarrow 0$
    $w_1, w_2 \leftarrow g[k], g[k + k_{\text{rel}}]$
    if $g[k] \neq (-1)$ **then**
      $v_1 \leftarrow \sum(g = g[k])$
    else
      $g[k] \leftarrow i$
      $v_1 \leftarrow 1$
    if $g[k + k_{rel}] \neq (-1)$ **then**
      $v_2 = \sum(g = g[k + k_{\text{rel}}])$
    else
      $g[k + k_{\text{rel}}] \leftarrow i$
      $v_2 \leftarrow 1$
  ▷ both groups unassigned:
    if $(v_1 = 1) \wedge (v_2 = 1)$ **then**
      $n[k + k_{\text{rel}}] \leftarrow n[k + k_{\text{rel}}] - n_{\text{tmp}}$
  ▷ first group bigger:
    if $(v_1 \geq v_2) \wedge (w_1 \neq w_2)$ **then**
      $n[g = w_1] = n[g = w_1] - n_{\text{tmp}}$
      $g[g = w_1] = g[w_2]$
  ▷ second group bigger:
    if $(v_1 < v_2) \wedge (w_1 \neq w_2)$ **then**
      $n[g = w_2] = n[g = w_2] + n_{\text{tmp}}$
      $g[g = w_2] = g[w_1]$
  ▷ Assign shifted phase:
  $\tilde{\varphi}_{\text{out}} =$ 0-array with size of $\tilde{\varphi}$
  for $i_x, i_y, i_z \in [0 : c_x, 0 : c_y, 0 : c_z]$ **do**
    $\tilde{\varphi}_{\text{out}}[a[i_x i_y i_z] : b[i_x i_y i_z]] \leftarrow \tilde{\varphi}[a[i_x i_y i_z] : b[i_x i_y i_z]] + 2\pi \cdot n[i_x i_y i_z]$
  return $\tilde{\varphi}_{\text{out}}$

where 'surf' computes the connectivity quality measure as in Eq. 6.

erated:

$$\varphi = \measuredangle \exp^{2i \cdot \varphi_{TE2}}$$

The brain mask was generated manually, utilising region growing with ITKSnap (Yushkevich et al., 2006).

According to the volume dimensions, subset sizes of $2r \times 2r \times r$ with $r = [16, 24, \ldots, 80]$ were chosen. A comparison of performance over 30 averages was assessed, by running AR on one and URSULA on 1, 2, 4 and 6 nodes. The required computing time was recorded and the results were compared for differences using the formula in Eq. (16) with the AR as reference.

3. Results

3.1. Simulation

For all averages, mean values of the computing time and median values of the error count were determined. The standard deviation was calculated as a measure for the distribution of the simulation results. However, a Gaussian distribution of the values was not expected.

On the hardware described above, AR could be computed up to a volume size of 304^3, while URSULA could be applied with up to 352^3 voxels due to memory restrictions.

The parameter plots in Fig. 6 illustrate the time minimum, implying the optimal subset dimensioning per volume size achievable with URSULA. The values are normalised to the minimum required time in each column (volume size). For computation on a single node, the optimum is mostly half of the volume size, meaning a subdivision factor of 8. For higher node counts, smaller subset dimensioning becomes advantageous above a certain volume size, e.g. from 128^3 at four nodes. For 4 and 6 computing nodes, the ideal computing time is reached by comparably small subset dimensioning.

Fig. 7 illustrates numerical results for the computing time, comparing the time-optimised URSULA subset dimensioning for each node count with the AR. URSULA clearly outperforms the AR for volume sizes bigger than 128^3 – even when only using one slave node. The Figure also shows the least-squares fit of $\alpha \cdot N^3$ for AR and $\alpha \cdot N^3 + \beta \cdot ls^2 + \gamma$ for URSULA, respectively. The fit describes the URSULA curve well for all node counts and also follows the timing results for AR below 256^3. Above this threshold, the AR requires substantially more time than expected.

Finally, Fig. 8 compares the result error of the algorithms for the application on six nodes, calculated by Eq. (16). The high node count was chosen to compare the average error of URSULA for very high parallelisation factors with that of AR.

For time-optimised subset dimensioning, the error is comparable between AR and URSULA, indicating only slightly higher values for URSULA. As a comparison, the highest occurring error (regarding subset dimensioning) is plotted. Compared with AR, the most adverse error level still remains in the same order of magnitude.

The behaviour of the result error within AR and URSULA for different noise levels is shown in Fig. 9. In order to consider both the realistic and adverse choice for the subset configuration, the setup with the fastest computing time and the setup with the highest error occurrence are compared. The error of the 'fastest' configuration correlates well with the behaviour of the AR approach and rises only slightly faster than AR regarding the noise level. For the 'highest error' configuration, the error is clearly higher than for the AR approach. Yet, it remains in the same order of magnitude and does not deviate greatly from the 'fastest'. Both algorithms provide good results up to an error level of about $\sigma = 0.4$ rad. Above this level, the result error increases exponentially.

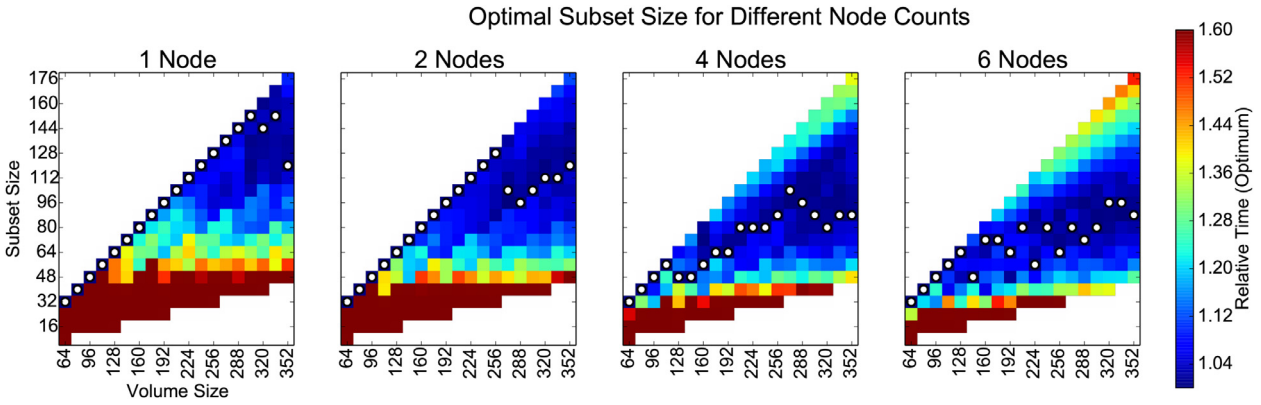

Fig. 6. Relative computing time required by URSULA for unwrapping the numeric sample (mean values over 30 averages) with $\sigma_\varphi = 0.25$ rad. In order to provide a global colour scale, each column (volume size) is normalised by its minimum computing time, which is indicated by a white dot. Hence, the normalised relative minimum per column always has a value of 1.0 and the other subset configurations along the vertical axis are given as relative values with respect to the minimum. (For interpretation of the full colour range, the reader is referred to the web version of this article.)

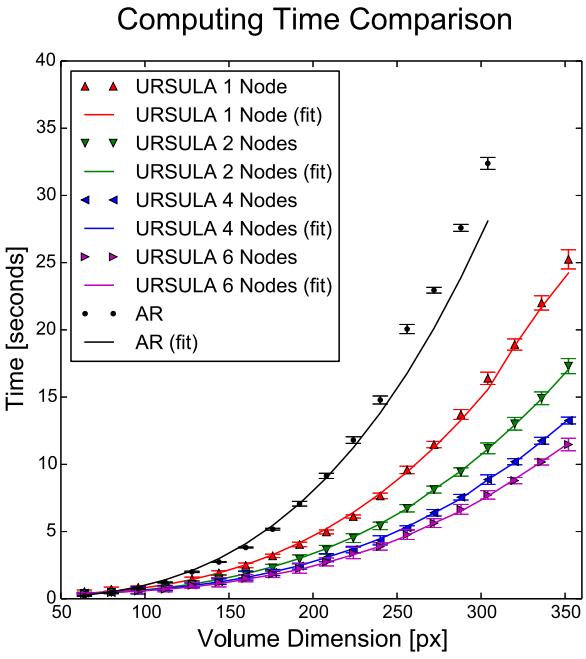

Fig. 7. Computing time (mean values over 30 averages) required by AR and URSULA (1, 2, 4 and 6 nodes) for simulated volumes of size N^3 (x axis) and $\sigma_\varphi = 0.25$ rad. Solid graphs indicate fitting results.

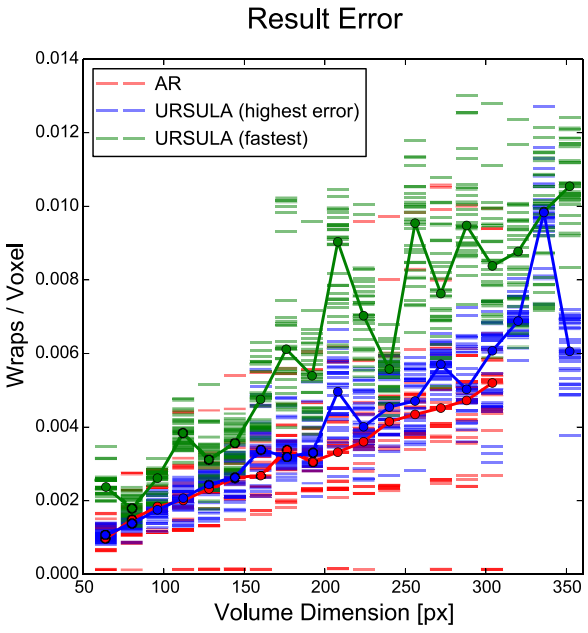

Fig. 8. Result error of AR and URSULA (6 nodes) for simulated volumes of size N^3 (x axis) and a noise level of $\sigma_\varphi = 0.25$ rad. Horizontal bars indicate individual simulations, whilst circles indicate median values. 'Fastest' refers to the subset dimensioning with fastest computing at 6 nodes, while 'highest error' refers to the dimensioning with highest error rate. (For interpretation of the coloured graphs, the reader is referred to the web version of this article.)

3.2. Measurement

Both algorithms produce satisfying unwrap results for the measured data as illustrated in Fig. 10. No significant wraps remain in the volume and the results appear widely similar. The figure shows characteristic unwrapping errors occurring in both algorithms, and an example of an URSULA-specific unwrapping error that can be caused by the volume subdivision scheme. These errors are mostly encountered near the mask surface and in regions with phase gradients close to the Nyquist limitation, as illustrated in Fig. 10.

The numerical difference between both results is quantified in Fig. 11. For higher subset dimensioning the difference reaches below 0.01 wraps/voxel. When referring to AR as the standard method, this agrees with the range of the error value expected in the simulation with comparable volume size.

The timing results for the volume of $310 \times 320 \times 360$ voxels are illustrated in Fig. 12 and listed in Table 1. Except for very small, non-optimised subset dimensioning, all URSULA configurations computed significantly faster than AR. The difference between 4 or 6 node computation is only marginal.

4. Discussion and perspective

The simulations show that moderate or large dimensioning of the subsets gives good results and needs to be optimised only for minimal time demands, whereas extremely low subset dimensions increase computing time. Adverse subset dimensioning, e.g. leading to fragmented mask parts, may occasionally increase the result error as seen in the *in vivo* measurement between subset dimensioning 40 and 48 (Fig. 11), but can easily be prevented by slightly increasing or decreasing the subset size. In general, URSULA reduces the computing time in the synthetic experiments by up to

Table 1
Computing time in seconds for AR and URSULA applied on the *in vivo* measurement over 30 averages.

		Subset dimensioning ($2n \times 2n \times n$) [voxel]								
	–	16	24	32	40	48	56	64	72	80
AR										
1 node	12.3874	–	–	–	–	–	–	–	–	–
$\pm\sigma$	0.218557	–	–	–	–	–	–	–	–	–
URSULA										
1 node	–	31.32	14.34	10.92	9.99	8.64	7.86	7.92	7.58	7.45
$\pm\sigma$	–	4.78	1.35	0.82	0.53	0.30	0.25	0.23	0.17	0.10
2 nodes	–	16.91	8.51	6.81	6.20	5.64	5.28	5.42	5.33	5.57
$\pm\sigma$	–	1.35	0.41	0.24	0.22	0.24	0.15	0.19	0.15	0.18
4 nodes	–	10.57	5.87	4.98	4.55	4.45	4.28	4.44	4.76	5.29
$\pm\sigma$	–	0.49	0.24	0.32	0.27	0.32	0.19	0.21	0.19	0.23
6 nodes	–	7.27	4.61	4.49	4.14	4.02	4.14	4.34	4.72	5.22
$\pm\sigma$	–	0.28	0.18	0.32	0.32	0.28	0.28	0.18	0.16	0.19

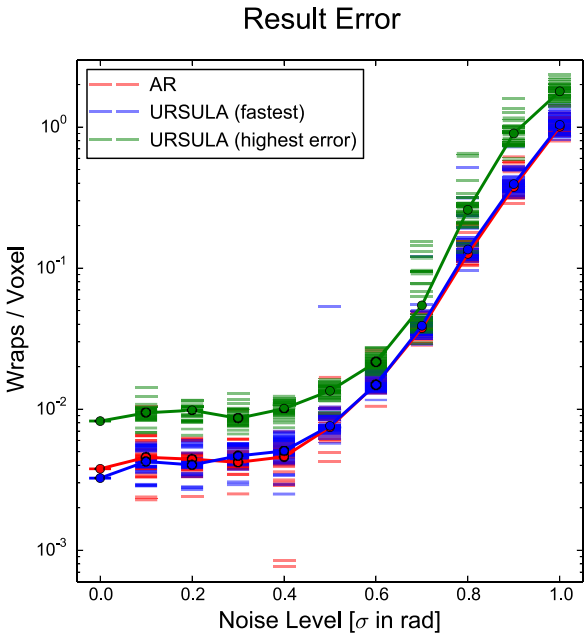

Fig. 9. Result error of AR and URSULA for different noise levels with a simulated volumes size of 256^3. Horizontal bars indicate individual simulations, whilst circles indicate median values. 'Fastest' refers to the subset dimensioning with fastest computing at 6 nodes, while 'highest error' refers to the dimensioning with highest error rate. (For interpretation of the coloured graphs, the reader is referred to the web version of this article.)

75%, in the presented *in vivo* case by more than 65% on the given hardware configuration.

This does not critically increase the error of the result. For time-optimal subset dimensioning, the error produced by URSULA in the simulation is close to that of the AR approach. With respect the AR unwrap error, URSULA reaches high deviations ($\leq$0.002 Wraps/Voxel) only for a few outliers. This is probably due to adverse subset and volume dimensioning. However, for high volume sizes the absolute error is still lower than 0.01 Wraps/Voxel. In order to achieve the lowest computing time for a given infrastructure, the subset dimensioning should, after setting an initial estimate, be optimised for volume size and for the number and computing power of the available parallel workers. The results show that time-optimised subset dimensions lead to result errors that remain very close to errors generated by the AR.

A few outliers of low error rate for the AR potentially originate in poorly connected regions that usually cannot be properly unwrapped but benefit from a spontaneous connection generated by the artificial noise or due to fortunate noise distribution. Such scenarios are much more unlikely using the subdivision in URSULA.

In the presented simulations, only volumes up to 352^3 voxels could be treated. The limiting factor is the memory available in the master node (8GB in the presented study) controlling the unwrapping process. This limitation causes the deviations in the timing curve of AR from $\alpha \cdot N^3$ at high volume sizes (see Fig. 7), as the (swap-free) system struggles with memory shortage and reallocations.

However, if larger volumes must be computed with URSULA, it is only necessary to increase the memory in the master node, whilst the slaves can be operated at lower memory with an according number of parallel processes.

In the simulation, the conditions for URSULA were optimised for different volume sizes within four computing node constellations. Yet, optimisation of the subset dimensions for a given volume is not only dependent on the number of workers and on the volume size, but is strongly influenced by the architecture and speed of the employed network as well as by the processors and memory of all involved workers. Hence, in order to achieve ideal results, the URSULA parameterisation must be optimised whenever it is introduced to a new computing network.

Interestingly, the result error per voxel seems to increase in a linear fashion with the volume size, although naively a constant error rate could be expected. Whilst taking into account that the rim of the evaluation mask usually coincides with regions of higher phase gradients, it is clear that unwrap errors are more likely to occur in this region. The wrap frequency per voxel is kept constant over the volume change for the artificial data, and hence the linear increase of the error with volume size appears more reasonable.

Furthermore, spikes in the error curve of URSULA (see Fig. 8) are related to the choice of the optimal subset dimensioning for time efficient computation. In the present study, best timing over best error rate was intentionally chosen in order to create a realistic impression. The strong peak in the 'fastest' curve at volume size $N = 336$ may therefore be caused by the choice of a small or adverse subset size.

Reduction of the computing time for single data sets enables new possibilities for instant on-line phase, field and susceptibility analysis. Although the absolute time gain of a few seconds seems moderate for current clinical standards, one has to direct the attention to the relative gain and future applications. The reduction of computing time by a factor of 3 or 4 per echo allows for faster evaluation and higher patient throughput at high field sites where individual receive channel evaluation becomes necessary. Scenarios in which the unwrap of significantly more than one volume becomes necessary, e.g. multi-channel receive (e.g. Robinson and Jovicich, 2011), multi-echo recording, and also iterative B_0 estimation, such as that required for shimming, will profit strongly.

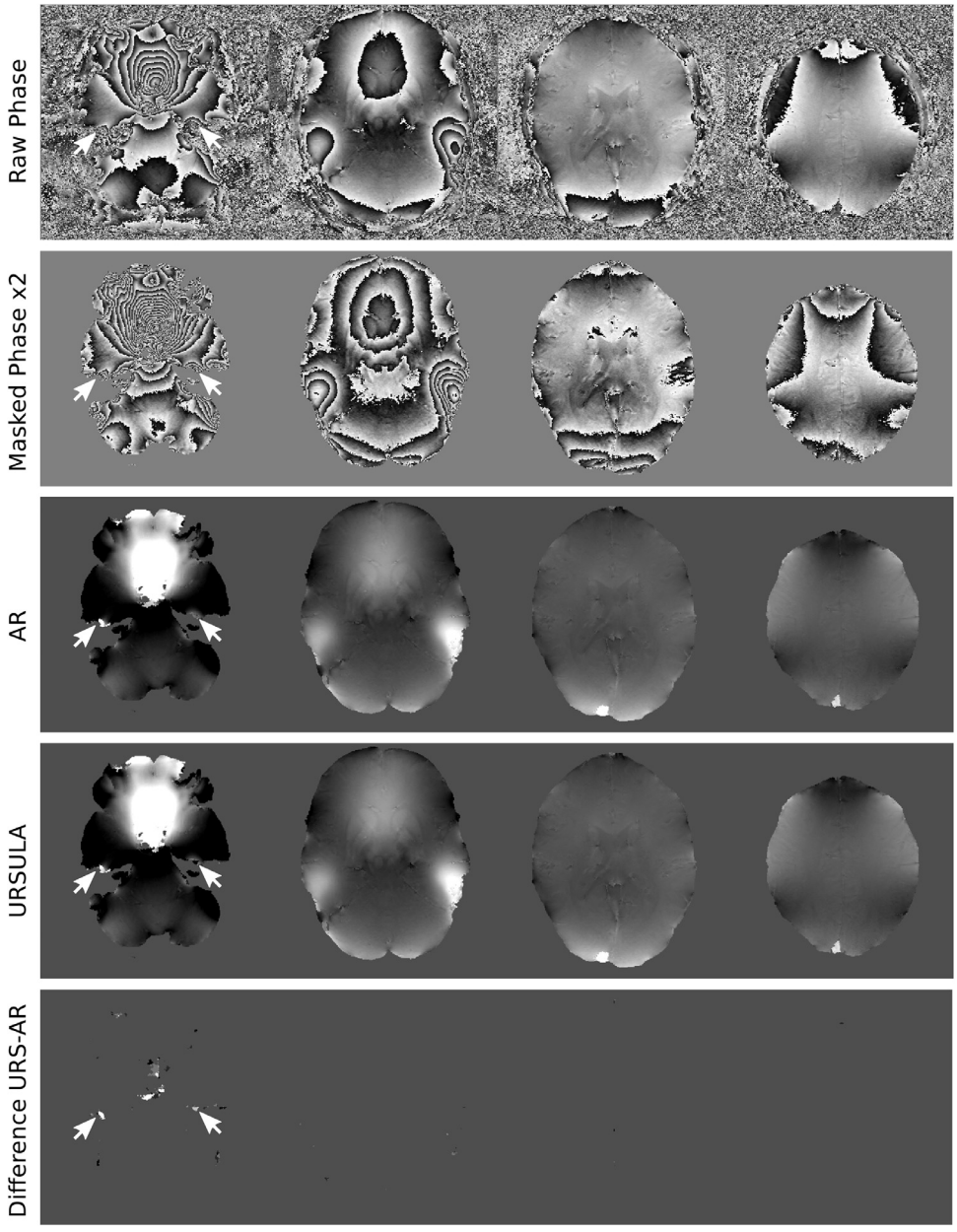

Fig. 10. Four transverse sample slices of the *in vivo* measurement, showing from top to bottom: raw phase, masked phase multiplied by a factor of two, AR, URSULA and the difference map of URSULA – AR. The leftmost slice is most inferior and shows a high number of wraps due to cavity-induced field distortions. The arrows on the left hand side indicate unwrap failures by both algorithms. The arrows on the right hand side indicate unwrap errors induced by the volume subdivision of URSULA.

As a numerical example, if a measurement with a resolution of 304^3, recorded over 16 echo time points with a 32-channel receive coil, and thus 512 volumes, is unwrapped, assuming the computing times from our simulations as a rough approximation, one can estimate the total computing times to be: $T_{\mathrm{AR}} = 512 \cdot 32 = 16\,384$ seconds and $T_{\mathrm{URSULA6}} = 512 \cdot 8 = 4\,096$ s. This represents a time gain of over 3 h 20 min.

Besides the obvious advantage in computing time savings, URSULA exhibits even more potential. The subdivision of the volume was chosen in Cartesian grid pattern – however, an arbitrary shape, e.g. guided by data quality, anatomy or mask shape is generally feasible. Furthermore, the parallelisation approach is applicable to other data processing tasks that evaluate the relation of neighbouring voxels.

The parallelisation technique, *dispy*, was chosen for the approach, since it is highly scalable. It can be set up on almost any constellation of workstations within one local area network. These do not have to be exclusively dedicated for URSULA processing, neither do they have to form a sophisticated computing cluster.

The presented algorithm performs very well within typical convex masks. Yet, a few pitfalls exist. A subset containing two disjoint regions may lead to an inconsistent unwrap result, as no mathematical relation is given inside of the subset. The same failure can occur if two regions within one subset are naturally connected over a pathway that lies partially outside of the subset volume – in this case the algorithm will choose the connection of highest quality available within the subset. However, the latter case is rare in brain imaging. Disjoint mask segments can be avoided by a slight change in subset dimensioning. Additionally, URSULA is subject to the accuracy of the employed subset unwrapping technique, which in the presented case is the AR. This means that the global unwrap of a subset with incorrectly unwrapped neighbours is prone to fail.

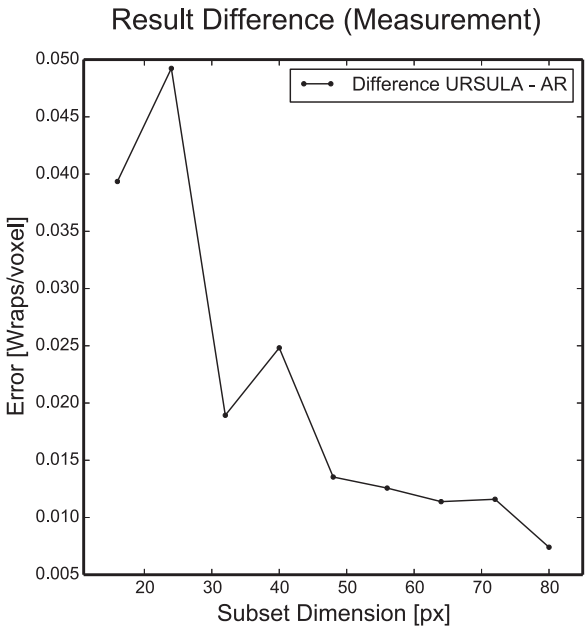

Fig. 11. Difference between URSULA and AR applied on the *in vivo* measurement in units of wraps/voxel

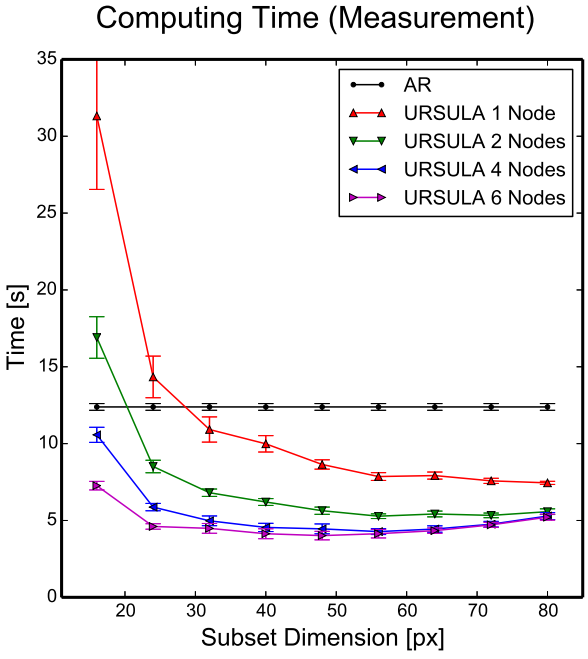

Fig. 12. Computing time required by AR (single node) and URSULA (1, 2, 4 and 6 nodes) with different subset dimensions. The AR timing result is repeated over all subset sizes for better visualisation. The plot shows the time average and standard deviation over 30 computed averages.

URSULA can easily be extended to include additional quality measures for subset-matching such as the average local phase gradient. In a further extension, subsets could be subdivided into individually computed disjoint mask segments in order to avoid the pitfalls described above.

5. Conclusions

The presented method, URSULA, enables unwrapping of three-dimensional phase data with four-fold (for the given computing network) or faster speed when compared to the underlying conservative AR algorithm. Data integrity and noise dependence remain in the same range and do not increase significantly when choosing subset dimensions that are time-optimised. This was successfully demonstrated on extensive simulations with known ground-truth as well as on a measurement example.

Given the same computing hardware, URSULA is able to handle larger data arrays than AR. Due to its conception, URSULA is able to compute extremely large arrays, requiring only the master node to provide higher memory resources.

URSULA embodies a universal way to parallelise a large-scale unwrapping problem by breaking it down into independent, small-scale problems. While the presented work features the AR algorithm, URSULA offers a free choice for the algorithm used to unwrap the subsets. It is even possible to use a combination of different algorithms.

We show that URSULA offers ideal properties to compute large, high resolution datasets of current and future acquisition techniques. Ultra-high field techniques such as phase and susceptibility mapping with multi-channel receive will profit greatly from the reduced computing times, taking yet another step towards real-time processing at the scanner site. Furthermore, restrictions to the volume size implied by the hardware are clearly reduced.

Aknowledgements

This work is funded in part by the Helmholtz Alliance ICEMED - Imaging and Curing Environmental Metabolic Diseases, through the Initiative and Network Fund of the Helmholtz Association. Further, our group is supported in part by TRIMAGE, an EU FP7 project (Grant Agreement N° 602621) and institutional funding.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.media.2018.11.004.

References

Abdul-Rahman, H.S., Gdeisat, M.a., Burton, D.R., Lalor, M.J., Lilley, F., Moore, C.J., 2007. Fast and robust three-dimensional best path phase unwrapping algorithm. Appl. Opt. 46 (26), 6623–6635.

Cusack, R., Papadakis, N., 2002. New robust 3-D phase unwrapping algorithms: application to magnetic field mapping and undistorting echoplanar images. Neuroimage 16 (3), 754–764. doi:10.1006/nimg.2002.1092.

Dagher, J., Reese, T., Bilgin, A., 2014. High-resolution, large dynamic range field map estimation. Magn. Reson. Med. 71 (1), 105–117. doi:10.1002/mrm.24636.

Deistung, A., Rauscher, A., Sedlacik, J., Stadler, J., Witoszynskyj, S., Reichenbach, J.R., 2008. Susceptibility weighted imaging at ultra high magnetic field strengths: theoretical considerations and experimental results. Magn. Reson. Med. 60 (5), 1155–1168. doi:10.1002/mrm.21754.

Dong, Y., Ji, J., 2010. Phase unwrapping using region-based markov random field model. Conf. Proc. IEEE Eng. Med. Biol. Soc. 2010, 3309–3312. doi:10.1109/IEMBS.2010.5627494.

Feng, W., Neelavalli, J., Haacke, E.M., 2012. Catalytic multiecho phase unwrapping scheme (CAMPUS) in multiecho gradient echo imaging: Removing phase wraps on a voxel-by-voxel basis. Magn. Reson. Med. 000 (January), 1–10. doi:10.1002/mrm.24457.

Goldstein, R.M., Zebker, H.a., Werner, C.L., 1988. Satellite radar interferometry: two-dimensional phase unwrapping. Radio Sci. 23 (4), 713. doi:10.1029/RS023i004p00713.

Haacke, E.M., Brown, R.W., Thompson, M.R., Venkatesan, M.R., 1999. Magnetic Resonance Imaging: Physical Principles and Sequence Design. Wiley-Liss.

Haacke, E.M., Liu, S., Buch, S., Zheng, W., Wu, D., Ye, Y., 2015. Quantitative susceptibility mapping : current status and future directions. Magn. Reson. Imaging 33 (1), 1–25. doi:10.1016/j.mri.2014.09.004.

Hedley, M., Rosenfeld, D., 1992. A new two-dimensional phase unwrapping algorithm for MRI images. Magn. Reson. Med. 24 (1), 177–181.

Jenkinson, M., 2003. Fast, automated, N-dimensional phase-unwrapping algorithm. Magn. Reson. Med. 49 (1), 193–197. doi:10.1002/mrm.10354.

Karout, S., Gdeisat, M.a., Burton, D.R., Lalor, M.J., 2007. Comparison between the residue-vector map and existing quality maps for phase unwrapping. J. Phys. Conf. Ser. 76, 012029. doi:10.1088/1742-6596/76/1/012029.

Li, W., Avram, A.V., Wu, B., Xiao, X., Liu, C., 2014. Integrated Laplacian-based phase unwrapping and background phase removal for quantitative susceptibility mapping. NMR Biomed. 27, 219–227. doi:10.1002/nbm.3056.

Lindemeyer, J., 2015. Optimisation of Phase Data Processing for Susceptibility Reconstruction in Magnetic Resonance Imaging, Phd. Thesis. Aachen, Techn. Hochsch., Aachen.

Lindemeyer, J., Oros-Peusquens, A.-M., Vahedipour, K., Shah, N.J., 2013. Quality-based UnwRap of SUbdivided Large Arrays (URSULA) at 9.4T. In: Proc. Int. Soc. Magn. Reson. Med., 21, p. 2494.

Liu, T., Liu, J., de Rochefort, L., Spincemaille, P., Khalidov, I., Ledoux, J.R., Wang, Y., 2011. Morphology enabled dipole inversion (MEDI) from a single-angle acquisition: comparison with COSMOS in human brain imaging. Magn. Reson. Med. 66 (3), 777–783. doi:10.1002/mrm.22816.

Liu, W., Tang, X., Ma, Y., Gao, J.-H., 2012. 3D phase unwrapping using global expected phase as a reference: application to MRI global shimming. Magn. Reson. Med. i, 1–9. doi:10.1002/mrm.24448.

Lotz, J., Meier, C., Leppert, A., Galanski, M., 2002. Cardiovascular flow measurement with phase-contrast MR imaging: basic facts and implementation. RadioGraphics 22, 651–671.

Reichenbach, J.R., 2012. The future of susceptibility contrast for assessment of anatomy and function. Neuroimage 62 (2), 1–5. doi:10.1016/j.neuroimage.2012.01.004.

Reichenbach, J.R., Venkatesan, R., Yablonskiy, D.a., Thompson, M.R., Lai, S., Haacke, E.M., Thompson, S.L., Haacke, E.M., 1997. Theory and application of static field inhomogeneity effects in gradient-echo imaging. J. Magn. Reson. Imaging 7 (2), 266–279.

Robinson, S., Jovicich, J., 2011. B0 mapping with multi-channel RF coils at high field. Magn. Reson. Med. 66 (4), 976–988. doi:10.1002/mrm.22879.

Schweser, F., Deistung, A., Lehr, B.W., Reichenbach, J.R., 2011. Quantitative imaging of intrinsic magnetic tissue properties using MRI signal phase: an approach to in vivo brain iron metabolism? Neuroimage 54 (4), 2789–2807. doi:10.1016/j.neuroimage.2010.10.070.

Strand, J., Taxt, T., 2002. Two-dimensional phase unwrapping using robust derivative estimation and adaptive integration. IEEE Trans. Image Process. 11 (10), 1192–1200. doi:10.1109/TIP.2002.804567.

Strand, J., Taxt, T., Jain, a.K., 1999. Two-dimensional phase unwrapping using a block least-squares method. IEEE Trans. Image Process. 8 (3), 375–386. doi:10.1109/83.748892.

Wang, Y., Liu, T., 2015. Quantitative susceptibility mapping (QSM): decoding MRI data for a tissue magnetic biomarker. Magn. Reson. Med. 73, 82–101. doi:10.1002/mrm.25358.

Windischberger, C., Robinson, S., Rauscher, A., Barth, M., Moser, E., 2004. Robust field map generation using a triple-echo acquisition. J. Magn. Reson. Imaging 20 (4), 730–734. doi:10.1002/jmri.20158.

Witoszynskyj, S., Rauscher, A., Reichenbach, J.R., Barth, M., 2009. Phase unwrapping of MR images using Phi UN – a fast and robust region growing algorithm. Med. Image Anal. 13 (2), 257–268. doi:10.1016/j.media.2008.10.004.

Yushkevich, P.A., Piven, J., Cody Hazlett, H., Gimpel Smith, R., Ho, S., Gee, J.C., Gerig, G., 2006. User-guided 3D active contour segmentation of anatomical structures: s ignificantly improved efficiency and reliability. Neuroimage 31 (3), 1116–1128.