LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-831311

Weighted Minimum Norm Algorithm for Improved Phase Unwrapping

T. K. Lakshmanan, K. M. Champley, K. A. Mohan

January 31, 2022

Electronic Imaging
San Francisco, CA, United States
January 17, 2022 through January 17, 2022

Disclaimer

Weighted Minimum Norm Algorithm for Improved Phase Unwrapping

Tegan F. Lakshmanan[1], Kyle M. Champley[2], K. Aditya Mohan[2]; [1]University of California, Berkeley; Berkeley, CA; [2]Lawrence Livermore National Laboratory; Livermore, CA.

Abstract

Phase unwrapping is an integral part of multiple imaging techniques, and as a result, a wide range of algorithms have been created to unwrap phases. One such algorithm is the minimum L^p-norm phase unwrapping algorithm. This algorithm transforms the phase unwrapping problem into a minimization problem of a certain functional, which it solves with an iterative method. However, the problem is usually not convex, and when there are many sharp edges in the data to be unwrapped, the algorithm often produces a local minimum with new discontinuities in originally smooth areas. To prioritize solutions which minimize the functional better in smooth areas, we use weights to deprioritize data lying along edges in the ground-truth image. This requires a method to find ground-truth edges using the wrapped image, which we describe. When using the modified algorithm, we generally obtain improved results on images with multiple edges (both lower errors and more correct edge placement).

Introduction

Many imaging technologies, including synthetic aperture radar (SAR) and x-ray phase contrast tomography, return a 2d grid of wrapped phase values rather than actual phase values.[1]-[5] This means we only know the phase values modulo 2π, lying within $(-\pi, \pi]$.[1][5] To produce accurate images, it is essential to unwrap the wrapped phase data – to recover the actual phase values from the wrapped ones.

At base, this problem is impossible. For an entry x in a grid of wrapped values, the unwrapped value could be $x + 2\pi k$ for any integer k. To unwrap the grid, we must make some assumptions. In general, we assume that the image is mostly smooth; few adjacent pixels will differ from each other by more than π.[1]-[3]

With this assumption, one way to proceed is to start at a specific pixel, take an adjacent pixel, and add/subtract a multiple of 2π so that the adjacent pixel is less than π away from the first one.[1]-[3] Next take a pixel adjacent to the second pixel and repeat the process; continue following a path around the entire image.[1]-[3] This is known as a path-following algorithm.[1][4][5] The problem is that some images contain residues, or cycles of pixels such that when traveling along the cycle and repeatedly adding the wrapped difference between the current and previous pixel, the resulting sum is nonzero.[1]-[3] In images with residues, choosing different paths leads to different unwrapped results.[1]-[3] Since most images that need to be unwrapped contain residues, this approach has been modified several times to produce more useful algorithms.

For example, Goldstein et al.'s branch cut method locates residues, places branch cuts connecting residues in specific ways, and unwraps along a path that does not cross any branch cuts.[2]-[4] Another approach, Flynn's minimum discontinuity method, repeatedly divides the image into two regions, adding 2π to one region to minimize the discontinuities in the image.[4]

Another type of phase unwrapping algorithm treats the problem like a minimization problem. Such minimum norm methods aim to find a solution with local gradients as similar as possible to wrapped gradients of the wrapped image (these are always between $-\pi$ and π), so they work by minimizing the error between the unwrapped gradients and the wrapped gradients of the wrapped image.[1][4] One algorithm in this category is the minimum L^p-norm algorithm,[1] which will be described in more detail below because it is the algorithm we have modified.

Our contributions to the phase unwrapping problem are:

- A modification to the minimum L^p-norm algorithm that deprioritizes ground-truth edges in the unwrapping process
- A method to find ground-truth edges using only the wrapped image

Background

Minimum L^p-norm Algorithm

Assume we have an $M \times N$ grid of wrapped phase values $\psi_{i,j}$ from which we hope to obtain a grid of unwrapped values $\phi_{i,j}$. We assume that the unwrapped solution is largely smooth; in other words, the local phase gradients (differences between adjacent pixels) are less than π.[1] Specifically, let W be the wrapping operator, which wraps pixels into $(-\pi, \pi]$, and

$$f_{i,j} = W(\psi_{i+1,j} - \psi_{i,j}) \tag{1}$$

$$g_{i,j} = W(\psi_{i,j+1} - \psi_{i,j}) \tag{2}$$

We assume that wherever possible, the unwrapped solution is such that $\phi_{i+1,j} - \phi_{i,j} = f_{i,j}$ and $\phi_{i,j+1} - \phi_{i,j} = g_{i,j}$.[1] This leads to the approach of minimizing the functional in (3), with $p \approx 0$.[1]

$$J = \sum_{i=0}^{M-2} \sum_{j=0}^{N-1} \left|\phi_{i+1,j} - \phi_{i,j} - f_{i,j}\right|^p + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} \left|\phi_{i,j+1} - \phi_{i,j} - g_{i,j}\right|^p \tag{3}$$

The result of setting the total variation of J equal to 0 is the following system of equations:[1]

$$(\phi_{i+1,j} - \phi_{i,j})U(i,j) + (\phi_{i,j+1} - \phi_{i,j})V(i,j) - (\phi_{i,j} - \phi_{i-1,j})U(i-1,j) + (\phi_{i,j} - \phi_{i,j-1})V(i,j-1) = c(i,j) \tag{4}$$

where

$$c(i,j) = f_{i,j}U(i,j) - f_{i-1,j}U(i-1,j) + g_{i,j}V(i,j) - g_{i,j-1}V(i,j-1) \quad (5)$$

$$U(i,j) = \begin{cases} \left|\phi_{i+1,j} - \phi_{i,j} - f_{i,j}\right|^{p-2} & if\ i < M-2, j < N-1 \\ 0\ otherwise \end{cases} \quad (6)$$

$$V(i,j) = \begin{cases} \left|\phi_{i,j+1} - \phi_{i,j} - g_{i,j}\right|^{p-2} & if\ i < M-1, j < N-2 \\ 0\ otherwise \end{cases} \quad (7)$$

The minimum L^p-norm algorithm solves this system iteratively, using the previous iteration's solution to find numerical values for $f_{i,j}$, $g_{i,j}$, $U(i,j)$, and $V(i,j)$ and approximate the system as a linear system.[1] During each outer iteration, a preconditioned conjugate gradient (PCG) method is used to solve the linear system.[1] This PCG method involves many inner iterations.[1] If an outer iteration produces a partially converged solution ϕ_l (where l is the outer iteration index) such that the wrapped residual $W(\psi - \phi_l)$ can be unwrapped uniquely, the algorithm ceases the iterations, unwraps $W(\psi - \phi_l)$ using some simple method, adds the result back to ϕ_l, and returns this.[1] In the implementation of the algorithm, $U(i,j)$ and $V(i,j)$ are normalized:[1]

$$U(i,j) = \frac{\varepsilon_0}{\left|\phi_{i+1,j} - \phi_{i,j} - f_{i,j}\right|^{2-p} + \varepsilon_0} \quad (8)$$

$$V(i,j) = \frac{\varepsilon_0}{\left|\phi_{i,j+1} - \phi_{i,j} - g_{i,j}\right|^{2-p} + \varepsilon_0} \quad (9)$$

It is recommended to set $\varepsilon_0 = 0.01$.[1]

The minimum L^p-norm algorithm is generally quite successful at solving phase unwrapping problems. However, when $p = 0$ (which is usually the case[1]), the problem is not convex or continuous, so the algorithm may produce a non-optimal local minimum. Especially when presented with a wrapped image where the ground-truth image contains several sharp edges, it often yields a sub-optimal solution with larger errors than usual and with extra edges in areas that are smooth in the ground-truth image.

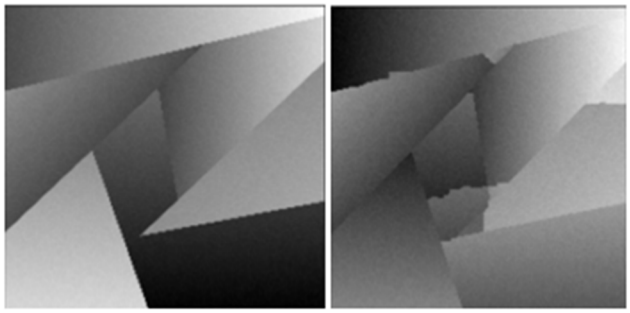

Figure 1. Example of the minimum L^p-norm algorithm's output when unwrapping an image with several sharp edges. (a) ground-truth image; (b) unwrapped image.

The Modification

The minimum L^p-norm algorithm struggles when the ground-truth image contains many edges because it prioritizes smoothness equally everywhere, including areas which are not actually smooth. As such, the algorithm may find a solution (a local minimum of the functional J) which is smooth in areas that should have edges and discontinuous in areas that should be smooth.

To mitigate this problem, we assign lower weights to terms in the functional J which contain pixels that are on edges in the ground-truth image. Weights have been used in other modifications to the minimum L^p-norm algorithm,[5] but to our knowledge they have not been used to de-prioritize ground-truth edges. Modifying the functional in this way forces the algorithm to prioritize smoothness in areas that are actually smooth. We let

$$J = \sum_{i=0}^{M-2}\sum_{j=0}^{N-1} \min(w_{i,j}, w_{i+1,j}) \left|\phi_{i+1,j} - \phi_{i,j} - f_{i,j}\right|^p + \sum_{i=0}^{M-1}\sum_{j=0}^{N-2} \min(w_{i,j}, w_{i,j+1}) \left|\phi_{i,j+1} - \phi_{i,j} - g_{i,j}\right|^p \quad (10)$$

where

$$w_{i,j} = \begin{cases} 1 \text{ if the pixel is not on an edge in the ground truth grid} \\ \alpha \text{ if the pixel is on an edge in the ground truth grid} \end{cases} \quad (11)$$

and $0 \leq \alpha < 1$. After testing several values of α, we found that $\alpha = 0.35$ provides a good compromise between speed and accuracy. We obtained the following results for other values of α:

- When $\alpha \approx 0$, the unwrapped image is very inaccurate in deprioritized areas
- For other values of $\alpha < 0.35$, the algorithm runs more slowly
- When $\alpha > 0.35$, there are often larger errors and extra edges in the unwrapped image

The effect of modifying the functional in this way is that the normalized versions of $U(i,j)$ and $V(i,j)$ are now

$$U(i,j) = \frac{\min(w_{i,j}, w_{i+1,j})\,\varepsilon_0}{\left|\phi_{i+1,j} - \phi_{i,j} - f_{i,j}\right|^{2-p} + \min(w_{i,j}, w_{i+1,j})\,\varepsilon_0} \quad (12)$$

$$V(i,j) = \frac{\min(w_{i,j}, w_{i,j+1})\,\varepsilon_0}{\left|\phi_{i,j+1} - \phi_{i,j} - g_{i,j}\right|^{2-p} + \min(w_{i,j}, w_{i,j+1})\,\varepsilon_0} \quad (13)$$

It is not obvious that we could find edges in the ground-truth image using only the wrapped image, so we will describe one solution to this problem below.

Finding Ground-truth Edges

Consider an edge in the wrapped image that is not present in the ground-truth image but rather caused by wrapping. Such an edge is characterized by pixels on one side with values near $-\pi$ and pixels on the other side with values near π. If we add some $\delta > 0$ to each pixel in the wrapped image, pixels on one side of the edge now have values near $-\pi + \delta$, and pixels on the other side now have values near $\pi + \delta$. When this new image is wrapped, the pixels with values near $\pi + \delta$ come to have values near $-\pi + \delta$.

Hence, the new wrapped image is continuous where the original wrapped image had an edge.

On the other hand, if an edge in the wrapped image corresponds to an edge in the ground-truth image, as long as the pixels across the edge are not uniformly 2π apart, adding δ throughout the image will not remove the edge.

Thus, to find ground-truth edges, we can run an edge-detection algorithm on the wrapped image, add some $\delta > 0$ to the wrapped image, rewrap, run edge detection on this, and take the edges which appear in the same places in both images. This process is illustrated in Figure 2. If the images are noisy and the edge detection is imperfect, it may be helpful to add a few different values of δ to the image and take the edges which appear in all of the corresponding rewrapped images.

The Modified Algorithm

Presented below is a summary of the modified algorithm. For more details on the implementation of the minimum L^p-norm algorithm, one should consult Ghiglia & Romero's paper.[1]

algorithm $\phi \leftarrow \text{ModifiedMinNorm}[\psi, p, l_{max}, k_{max}]$
Run edge detection on the wrapped image.
Add δ throughout the wrapped image, rewrap, and run edge detection on the resulting image.
Define $w_{i,j}$ as in equation (11), assuming ground-truth edges are those that appear in the same places in both images.
Set $\phi_l = 0$.
for $l \leq l_{max}$ **do**
Compute $W(\psi - \phi_l)$ and test for residues.
if $W(\psi - \phi_l)$ contains residues **do**
Compute $U_l(i,j)$ and $V_l(i,j)$ from equations (12) and (13), using ϕ_l to obtain numerical values. Using these values, compute $c_l(i,j)$ from equation (5).
Solve system of equations (4) using Ghiglia & Romero's Algorithm WLS (a PCG algorithm) with k_{max} the maximum number of inner iterations.[1]
Update ϕ_l with the output of Algorithm WLS.
else
Unwrap $W(\psi - \phi_l)$ with a simple phase unwrapping algorithm. Add the result to ϕ_l.
$\phi \leftarrow \phi_l$. Stop.
end if-else
if the algorithm has converged **do**
$\phi \leftarrow \phi_l$. Stop.
end if
end for
end algorithm

Results

We present three test cases, where we compared the results of the modified and original algorithms (with $p = 0$) on wrapped images for which we know the ground-truth images. In each test, we measured the error in the unwrapped images in two ways. The first error measure, E_1, is calculated according to the following formula, where the ground-truth phase values are given by $\theta_{i,j}$:

$$E_1(i,j) = \begin{cases} 0 & \text{if the corresponding pixel is on an edge in the ground truth image} \\ \left|\left(\phi_{i+1,j} - \phi_{i,j}\right) - \left(\theta_{i+1,j} - \theta_{i,j}\right)\right| + \left|\left(\phi_{i,j+1} - \phi_{i,j}\right) - \left(\theta_{i,j+1} - \theta_{i,j}\right)\right| & \text{otherwise} \end{cases} \quad (14)$$

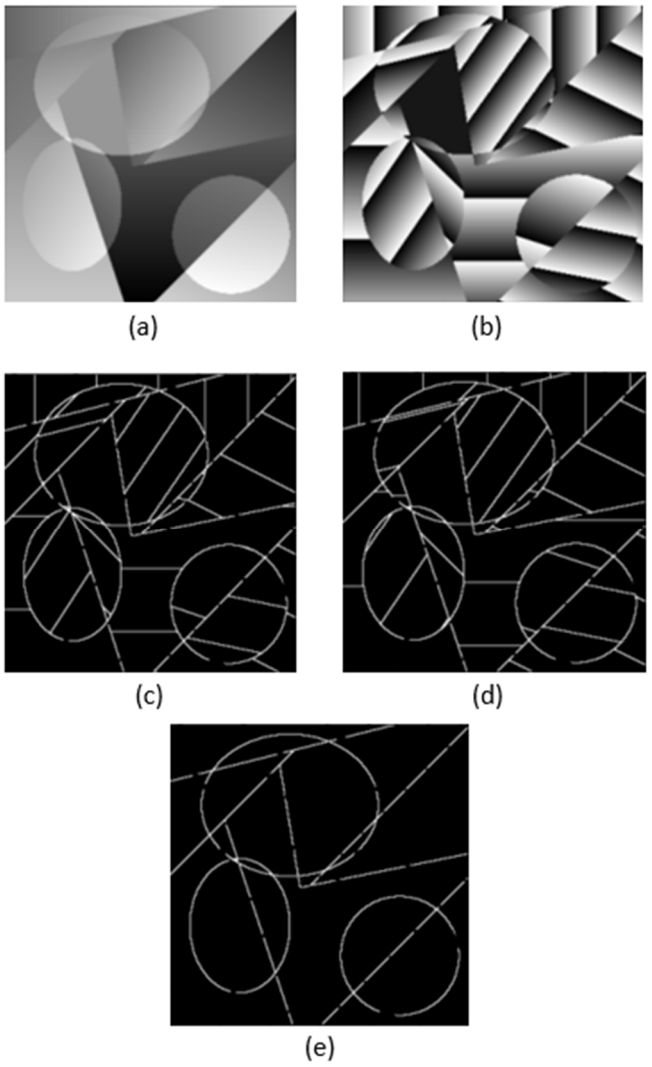

Figure 2. Illustration of the edge-finding process. (a) ground-truth image; (b) wrapped image; (c) edges of wrapped image; (d) edges of the image obtained by adding $\delta = 1.5$ throughout the wrapped image and rewrapping; (e) edges common to (c) and (d). Note that these are precisely the edges of the ground-truth image.

E_1 measures the error in local phase gradients of the unwrapped image in areas that are smooth in the ground-truth image. When calculating the average E_1 value of an image, we only average over areas that are smooth in the ground-truth image.

The second measure, E_2, is given by the following formula:

$$E_2(i,j) = \left|W\left(\left(\phi_{i+1,j} - \phi_{i,j}\right) - \left(\theta_{i+1,j} - \theta_{i,j}\right)\right)\right| + \left|W\left(\left(\phi_{i,j+1} - \phi_{i,j}\right) - \left(\theta_{i,j+1} - \theta_{i,j}\right)\right)\right| \quad (15)$$

E_2 measures the wrapped error in local phase gradients throughout the image. We wrap the error because in general, we cannot expect either the original minimum L^p-norm algorithm or the modified algorithm to produce results with accurate phase gradients across sharp edges. The best that we can hope for is that these phase gradients are off by multiples of 2π.

The algorithms were written in Python and timed on a laptop with an Intel i7 processor, 2.8 GHz processor speed, and 16GB of RAM. Because our method of finding ground-truth edges requires edge detection on wrapped images, and the focus of this project was not to find an accurate edge-detection method for noisy wrapped images, we used our knowledge of the ground-truth edges in the noisy test cases. In the noiseless test, we used the method of finding ground-truth edges described in this paper, with a very simple algorithm to find wrapped edges. We consider each algorithm to have converged if the value of the functional J has not changed in four successive outer iterations.

The first set of results is shown in Figure 3. The ground-truth image is a 200x200 pixel noiseless image with few edges, which the original algorithm performed well on. The goal was to test how the modified algorithm's performance would compare.

In Figure 3, we see that the unwrapped results of the modified and original algorithms look identical. The error plots in Figure 4 reveal that the modified method's result has more accurate phase gradients. It has average E_1 and E_2 values almost three times lower than the original algorithm's output, as shown in Table 1. Both algorithms ran for approximately the same amount of time. Even in images with few edges, the modified algorithm often performs better than the original.

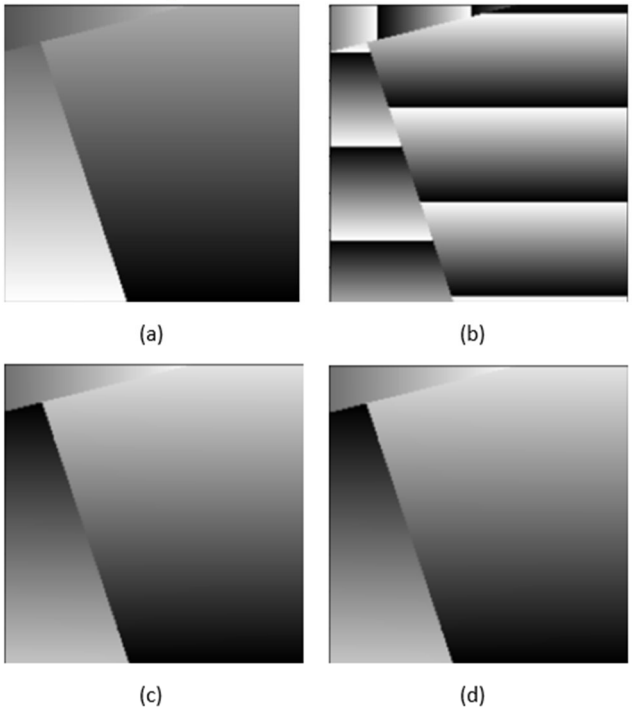

Figure 3. Comparison of the modified and original algorithm on the first test image. (a) ground-truth image; (b) wrapped image; (c) unwrapped image obtained from the original algorithm; (d) unwrapped image obtained from the modified algorithm.

The second set of results is shown in Figure 5. The ground-truth image is a noisy 200x200 image with several edges, which the original algorithm struggles to unwrap correctly. The goal is to test whether the modified algorithm will improve the results.

From Figure 5, we can see that the modified algorithm's result better reflects the structure of the ground-truth image. Table 2 also shows that the modified algorithm's result has an E_1 value almost twenty times lower than the original algorithm's result, and an E_2 value almost twice as low. Additionally, in this case, the modified algorithm is faster than the original.

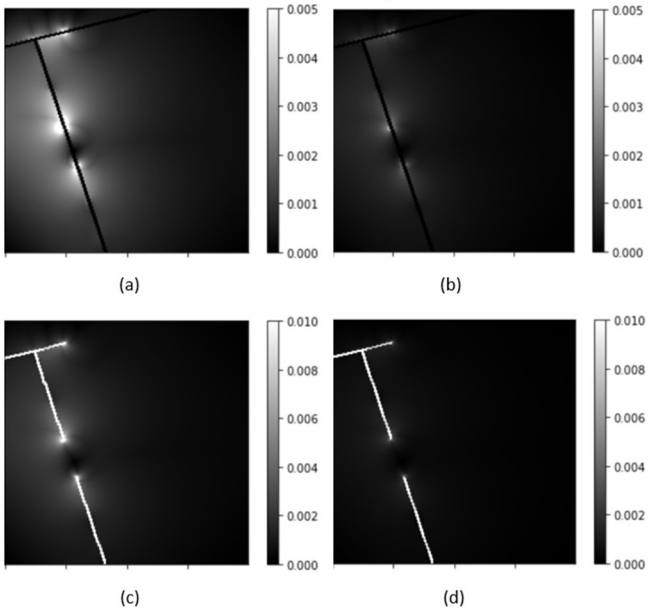

Figure 4. Comparison of the errors in the unwrapped images obtained via running the modified and original algorithm on the first test image. (a) plot of measure E_1 for the image from Figure 3c; (b) plot of measure E_1 for the image from Figure 3d; (c) plot of measure E_2 for the image from Figure 3c; (d) plot of measure E_2 for the image from Figure 3d.

Table 1: Average error measures in the unwrapped results from the first test image

	Unwrapped with Original Algorithm	Unwrapped with Modified Algorithm
Average E_1 Value	0.000978	0.000345
Average E_2 Value	0.00210	0.000739
Runtime	137 seconds	131 seconds

The third set of results is shown in Figure 7. The ground-truth image is another noisy 200x200 image with several edges. Again, the original algorithm's result has structural issues.

In the previous tests, both algorithms converged before reaching a point where the residual $W(\psi - \phi_l)$ could be unwrapped uniquely. In this test, the original algorithm reached the residual stopping condition. When this happens, we unwrap $W(\psi - \phi_l)$ with a method from the Python package skimage. The result is extremely accurate wrapped phase gradients and near-zero E_2 values, as shown in Figure 8. The modified algorithm converged before reaching this stopping condition, meaning its E_2 values are much larger. Plus, it took significantly longer for the modified algorithm to run (287 seconds vs. 91). However, Table 3

shows that the modified algorithm's E_1 values are over ten times lower, and the structure of its unwrapped image is correct.

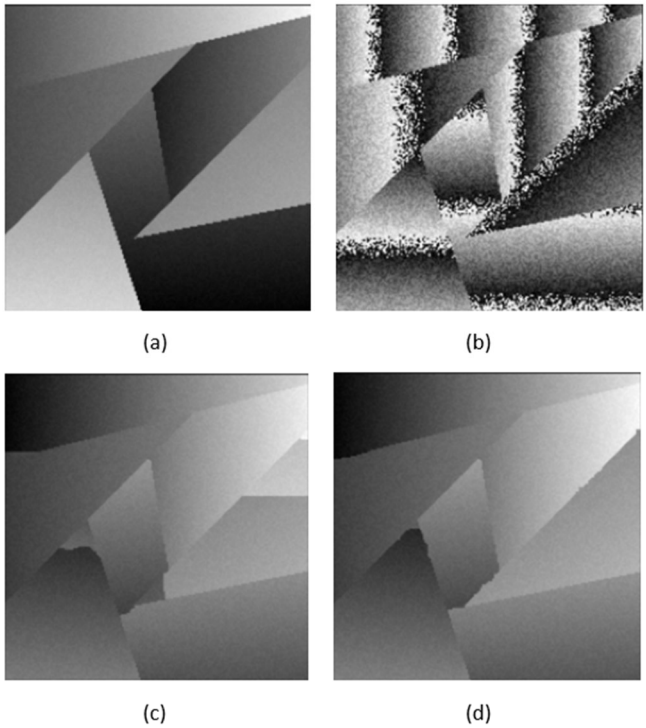

Figure 5. Comparison of the modified and original algorithm on the second test image. (a) ground-truth image; (b) wrapped image; (c) unwrapped image obtained from the original algorithm; (d) unwrapped image obtained from the modified algorithm.

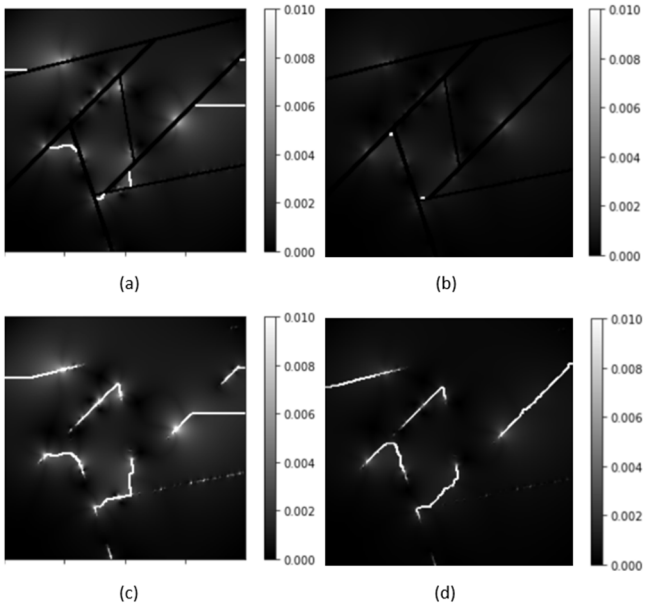

Figure 6. Comparison of the errors in the unwrapped images obtained via running the modified and original algorithm on the second test image. (a) plot of measure E_1 for the image from Figure 5c; (b) plot of measure E_1 for the image from Figure 5d; (c) plot of measure E_2 for the image from Figure 5c; (d) plot of measure E_2 for the image from Figure 5d.

Table 2: Average error measures in the unwrapped results from the second test image

	Unwrapped with Original Algorithm	Unwrapped with Modified Algorithm
Average E_1 Value	0.0201	0.00117
Average E_2 Value	0.00202	0.00108
Runtime	321 seconds	261 seconds

This demonstrates that occasionally, there are trade-offs between the modified and original algorithms. In our experience, it is usually the case that when the original algorithm reaches a point where $W(\psi - \phi_l)$ can be unwrapped uniquely, the modified algorithm also does. But on occasion, the modified algorithm converges instead. In these cases, the modified algorithm produces higher E_2 values and may run more slowly, but it also results in a more accurate image structure and lower E_1 values. It may be possible to make these cases even rarer by strengthening the condition for convergence. It should also be noted that on occasion, the modified algorithm reaches the residual stopping condition when the original algorithm does not, leading to marked improvement with the modified algorithm.

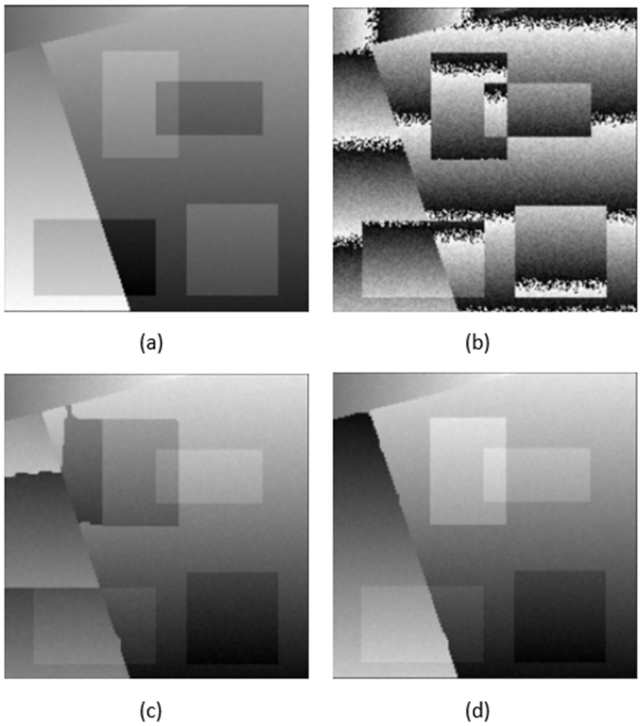

Figure 7. Comparison of the modified and original algorithm on the third test image. (a) ground-truth image; (b) wrapped image; (c) unwrapped image obtained from the original algorithm; (d) unwrapped image obtained from the modified algorithm.

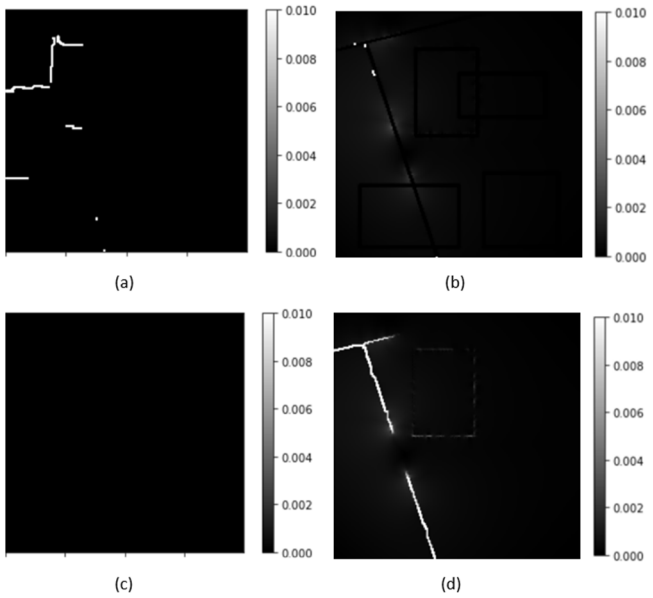

Figure 8. Comparison of the errors in the unwrapped images obtained via running the modified and original algorithm on the third test image. (a) plot of measure E_1 for the image from Figure 7c; (b) plot of measure E_1 for the image from Figure 7d; (c) plot of measure E_2 for the image from Figure 7c; (d) plot of measure E_2 for the image from Figure 7d. While the E_2 values produced by the modified method are higher in this case, the E_1 values are lower and the overall image structure is more accurate.

Table 3: Average error measures in the unwrapped results from the third test image

	Unwrapped with Original Algorithm	Unwrapped with Modified Algorithm
Average E_1 Value	0.0340	0.00273
Average E_2 Value	2.60×10^{-16}	0.000683
Runtime	91 seconds	287 seconds

Conclusion

We have presented a modification to the minimum L^p-norm phase unwrapping algorithm that involves finding edges in the ground-truth image and weighting these parts of the image lower in the unwrapping process. As part of this, we have described a method of finding edges in the ground-truth image using only the wrapped image. After finding these edges, the modification requires very few adjustments to the original algorithm.

When testing the performance of the modified algorithm, we found encouraging results. In test cases with few edges, where the original algorithm already performs well, the modified algorithm produces results with lower errors and the same correct structure (Figures 3 and 4). In test cases with more edges, where the original algorithm produces images with structural problems, the modified algorithm generally produces much improved results, with lower errors and more correct overall structures (Figures 5 and 6). In rare cases, there are trade-offs between the modified and original algorithms, and one must choose between a more accurate image structure and lower wrapped phase gradient errors (Figures 7 and 8). However, on the whole, the results from the modified method are very promising.

References

[1] D. Ghiglia & L. Romero, "Minimum L^p-Norm Two-Dimensional Phase Unwrapping," *Jour. Opt. Soc. Am. A*, vol. 13, no. 10, pp. 1999-2013.

[2] H. Zebker & Y. Lu, "Phase Unwrapping Algorithms for Radar Interferometry: Residue-cut, Least-squares, and Synthesis Algorithms," *Jour. Opt. Soc. Am. A*, vol. 15, no. 3, pp. 586-598.

[3] R. Goldstein, H. Zebker & C. Werner, "Satellite Radar Interferometry: Two-dimensional Phase Unwrapping," *Radio Science*, vol. 23, no. 4, pp. 713-720.

[4] S. Heshmat, S. Tomioka & S. Nishiyama, "Performance Evaluation of Phase Unwrapping Algorithms for Noisy Phase Measurements," *International Jour. of Optomechatronics*, vol. 8, no. 4, pp. 260-274.

[5] Y. Lu & X. Zhang, "Minimum L^0-Norm Two-Dimensional Phase Unwrapping Algorithm Based on the Derivative Variance Correlation Map," *Jour. Phys.: Conf. Ser.*, vol. 48, no. 57, pp. 308-312.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52- 07NA27344, document number LLNL-CONF-831311.

Author Biography

Tegan Lakshmanan (lakshmanan2@llnl.gov) is a fourth-year undergraduate student at UC Berkeley studying math and linguistics. She spent the summer of 2021 working with Lawrence Livermore National Lab on computational imaging research, and she plans to begin pursuing a math PhD in the next year.

Kyle Champley, PhD: Kyle joined LLNL in 2012 and is an applied mathematician who works in the Nondestructive Characterization Institute and is the lead for the Signal and Image Processing Research Group. He develops algorithms and writes software for data processing and reconstruction of Computed Tomography (CT) data. Kyle is the primary developer for the Livermore Tomography Tools (LTT) software package. He currently serves as a scientific consultant for three medical imaging startup companies. Previously, he was a staff scientist at the General Electric Global Research center where he developed CT reconstruction algorithms for GE's Revolution CT system. Kyle received his Ph.D. in Electrical Engineering from the University of Washington where he performed research in Positron Emission Tomography (PET).

K. Aditya Mohan (mohan3@llnl.gov) received his Ph.D. degree in electrical and computer engineering from Purdue University in 2017. He is affiliated with the Computational Engineering Division at Lawrence Livermore National Laboratory, Livermore, California, 94551, USA. His research interests include computational imaging, inverse problems, and machine learning. As a principal investigator, he is also experienced in leading research projects in these areas. He is a Senior Member of IEEE.