

Phase-unwrapping algorithm based on an adaptive criterion

J. A. Quiroga, A. González-Cano, and E. Bernabeu

A new algorithm for phase unwrapping of phase maps with noise or logical inconsistencies is proposed. It is based on the use of an adaptive threshold and the second difference of the locally unwrapped phase as a selection criterion for the pixels to be processed.

Key words: Phase unwrapping, fringe-pattern processing.

Phase-unwrapping algorithms must include procedures to handle problems such as noise or abrupt fringe breaks to achieve a final error-free continuous phase map. In general, points in the phase map that present these types of problems must not be processed, and its phase must be inferred from the resultant continuous phase map. However, sometimes it is difficult to identify all the conflictive points. In these cases the points that are suspected to be conflictive must be, at the least, the last to be processed, if possible, to avoid error propagation across the continuous phase map. In recent years, several methods have been proposed to do this. These methods choose different criteria to detect the conflictive points: modulation,^{1,2} first difference of phase, or fringe density.^{3,4}

Bone⁵ has proposed a method based on the second difference of the locally unwrapped phase in which the conflictive points are selected by comparison of the value of the second difference with a fixed threshold. The choose of this criterion permits one to detect phase discontinuities that produce integer fringe shifts, which is not possible with the use of a first-difference-based criterion. Therefore second-difference-based methods are well suited for phase maps with such problems.

In this note we present an algorithm based on the second difference of the locally unwrapped phase that uses an adaptive threshold instead of a fixed one. Some of its main features have been used in a previous study of the authors.³ In Ref. 3 a queue was used for the processing of arbitrary shapes, and a selection criterion, based in the phase gradient modulo 2π , determined which pixels were to be processed. The pixels in the neighborhood of a given pixel were ordered according to this criterion, and a fixed number of them were rejected. In this way the direction of lowest gradient modulo 2π was the preferred one for the progression of the phase-unwrapping algorithm. The rejection of a certain number of pixels in every step guaranteed that the pixels with the highest probability of being invalid ones were not processed at all or were at least the last processed ones. In this

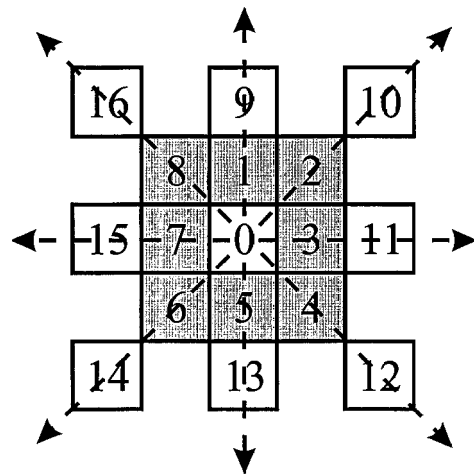


Fig. 1. Neighbor set for a given pixel, 0, and directions for the calculation of the second differences.

J. A. Quiroga and E. Bernabeu are with the Departamento de Óptica, Facultad de Ciencias Físicas, Universidad Complutense, Ciudad Universitaria, Madrid 28040, Spain. A. González-Cano is with the Escuela Universitaria de Óptica, Universidad Complutense, Arcos de Jalón, Madrid 28037, Spain.

Received 6 September 1994; revised manuscript received 17 January 1995.

0003-6935/95/142560-04\$06.00/0.

© 1995 Optical Society of America.

way, propagation of errors in the unwrapping process was stopped or at least limited.

The main disadvantages of this algorithm were the following:

(1) The fact that the number of points to be rejected is fixed implies that too much valid information can be rejected in noise-free zones of the phase map, whereas too much nonvalid information is considered in noisy zones.

(2) The use of local phase gradient modulo 2π as a selection criterion makes it difficult to detect the

zones of high curvature of the phase map that are usually associated with logical inconsistencies or discontinuities resulting in integer fringe shifts.

The algorithm that we present here is designed to deal with these types of problems. As in the previously depicted algorithm, the processing begins with any starting point that becomes the first so-called integrator pixel. We consider the 16 neighbors of this pixel represented in Fig. 1, and we calculate the second difference of the locally unwrapped phase in the directions defined by the eight connected neigh-

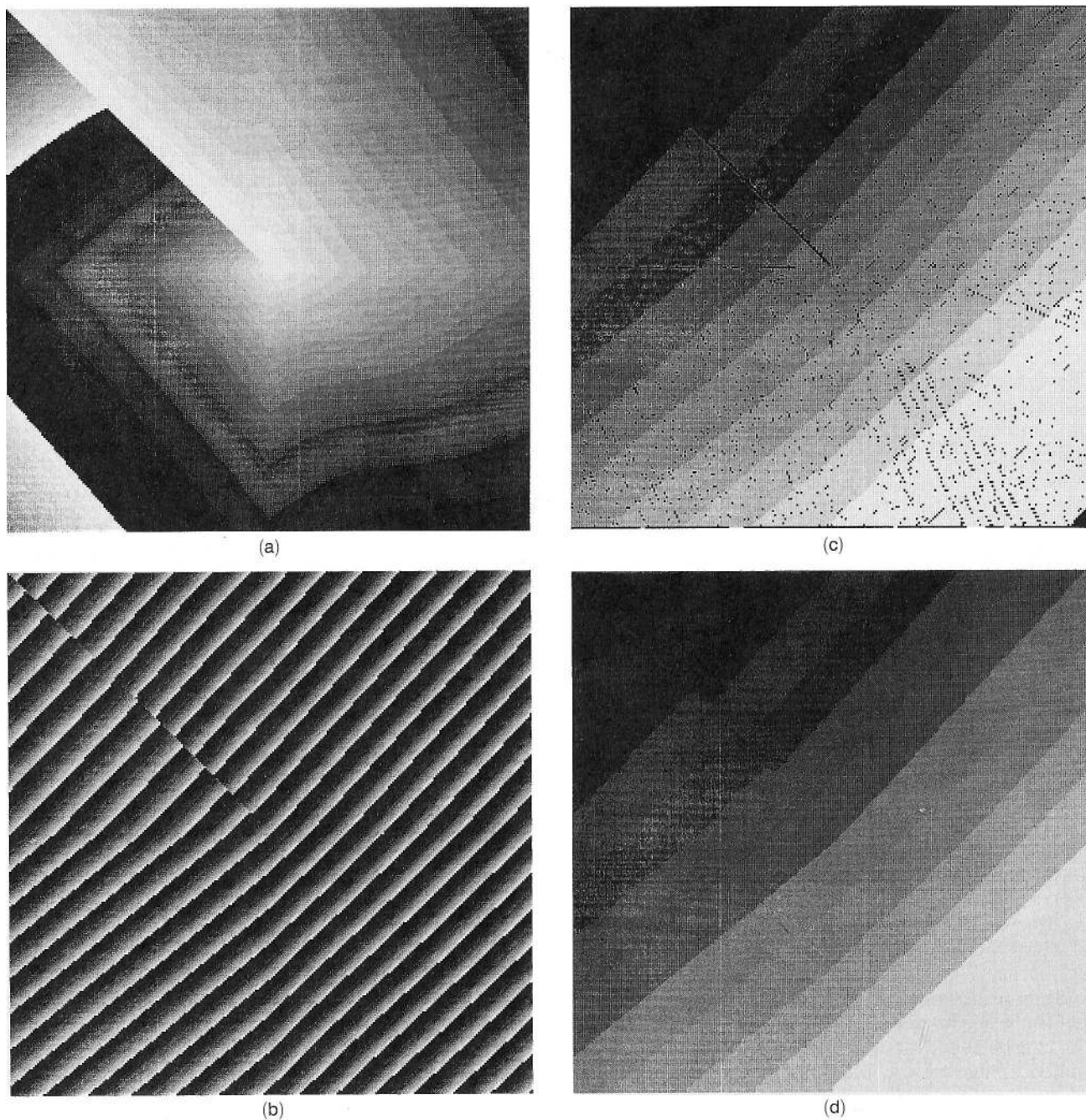


Fig. 2. (a) Computer-generated phase map with a discontinuity that produces an integer fringe shift. (b) Phase map of (a) when a linear carrier phase is added. (c) Unwrapped phase map when the second-difference-based algorithm is used. Black points correspond to nonprocessed points. (d) Final unwrapped phase map, when the value of the phase in the nonprocessed points is obtained by interpolation.

bors 1–8. For instance, if we consider the direction defined by pixels 0 and 3, this second difference is given by

$$\Delta(3) = 2\Phi^*(3) - \Phi^*(0) - \Phi^*(11), \quad (1)$$

where Φ^* denotes the locally unwrapped phase. We can calculate Φ^* for the pixels in this direction by⁴

$$\begin{aligned} \Phi^*(0) &= \phi(0), \\ \Phi^*(3) &= \Phi^*(0) + [\phi(3) - \phi(0)], \\ \Phi^*(11) &= \Phi^*(3) + [\phi(11) - \phi(3)], \end{aligned} \quad (2)$$

where $[\]$ denotes modulo 2π and $\phi(i)$ is the wrapped phase of pixel i . Equation (1) may then be written as

$$\Delta(3) = [\phi(3) - \phi(0)] - [\phi(11) - \phi(3)]. \quad (3)$$

We assign to each pixel the value of the second difference in the direction defined by the integrator pixel and itself. We order the pixels 1–8 according to the absolute values of their second difference, and we store them in the queue to be processed. Then, the first pixel in the queue (let us denote it by p) is extracted, and we calculate its unwrapped phase, Φ , with respect to the integrator pixel, 0, by

$$\Phi(p) = \Phi(0) + [\phi(p) - \phi(0)]. \quad (4)$$

Now, we consider p as a new integrator pixel and proceed again in the way depicted above, ordering its eight connected neighbors according to their second differences. However, not all the neighbors of p are stored. We define a threshold T based on the values of the second differences of the neighbors of the first integrator pixel, 0. A neighbor of p is rejected and subsequently not stored in the queue if the value of the second difference for the direction defined by p and itself is strictly bigger than T . We may use as T , for instance, the third value in the list of second differences for pixel 0 (this is a good estimation for the median of the set of the eight second differences for 0); any other value, i.e., the second, the fourth, etc., could be used depending on the image. We found that the third is a good compromise between noise management and the size of the processed area of the phase map.

According to this criterion, a certain number of neighbors of p are rejected, and the remaining ones are stored in the queue. Then, we take the first pixel in the queue, and we calculate its unwrapped phase, as in Eq. (4). We take it as the new integrator pixel, and we recalculate the threshold, T , for the new set of neighbors. We repeat these operations (obviously avoiding processing of previously processed pixels) until there are no pixels in the queue.

As can be seen, instead of using a fixed number of rejected pixels, we use a threshold that may vary in each step. The variation of the threshold becomes imposed by the local characteristics of the phase map because the value of T is obtained from the behavior of the phase in a zone near the pixel in which the

unwrapped phase is calculated. This means that the algorithm adapts itself to the local characteristics of the area that is processed and thus it overcomes the problem of considering too many or too few pixels in zones with different amounts of noise. Also, the use of the second difference as the selection criterion instead of the local gradient modulo 2π lets one detect and avoid the areas of high curvature.

To illustrate the way that the algorithm works, we generated a simulated phase map that contains a discontinuity in the phase field [Fig. 2(a)]. Figure 2(b) shows the phase map when a linear carrier phase is added. As can be seen, the discontinuity reaches values bigger than 2π rad (one fringe), thus producing an integer fringe shift. The discontinuity can be detected with our algorithm even at the place in which an integer fringe shift appears because in this place there still exists a small change in the curvature of the phase. In this way we can produce an error-free continuous phase map, shown in Fig. 2(c). The black points of Fig. 2(c) correspond to non-processed points whose phase values can be interpolated from the values of their neighbors. The result of this

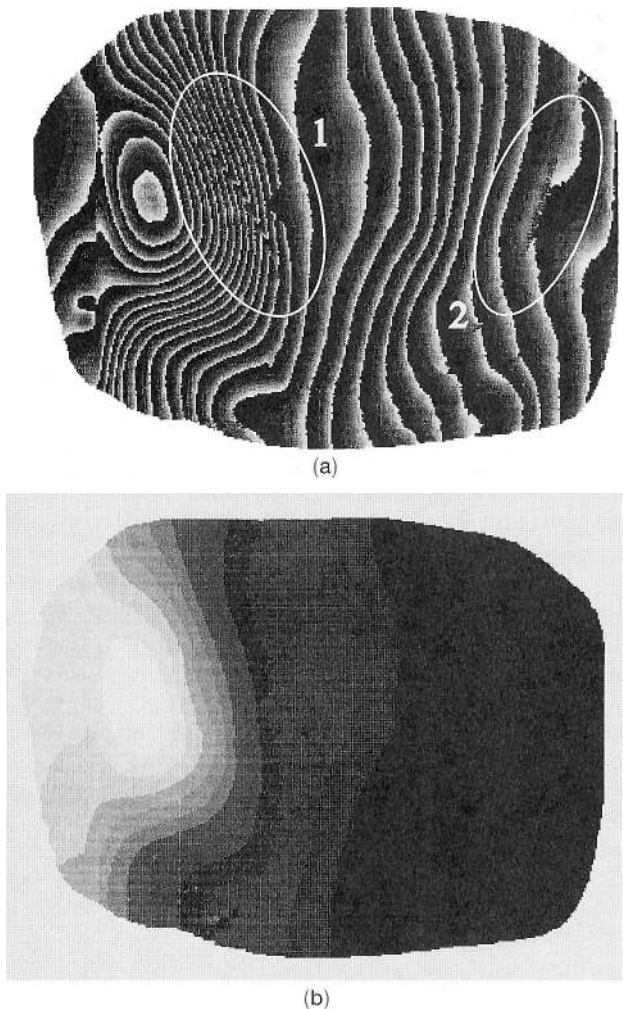


Fig. 3. (a) Phase map with two conflictive areas: 1, abrupt fringe ends and a very high curvature zone, 2, a 2π jump hidden by noise. (b) Final unwrapped phase map.

interpolation (i.e., the final unwrapped phase map) is shown in Fig. 2(d).

In a phase map with problems of the type discussed, an algorithm based on first differences is not effective because it cannot detect the discontinuity when it produces an integer fringe shift. Also, as Bone shows,⁵ a branch-cut method is not advisable because the number of positive and negative discontinuity sources is not the same in the conflictive region, which results in isolated discontinuity sources far from the boundary. This means that the isolated discontinuity sources must be joined to the border of the map, thus producing an incorrect distribution of cut lines. The computing time of our method was approximately 3 min. on a PC 486 66-MHz microcomputer for a size of 128×128 pixels. A branch-cut method followed by a simple unwrapping algorithm can be faster, but as we have said, it cannot be applied in this case. The election of our algorithm or a branch-cut method will then depend on the phase map being processed.

Figure 3 shows the results of the method in a real case. The phase map [Fig. 3(a)] presents two types of problems: a zone of high curvature (marked with 1) and a 2π jump hidden by noise (zone 2). As can be seen in the unwrapped phase map shown in Fig. 3(b), the algorithm can overcome both problems. The use of an adaptive threshold is efficient in cases such as this one, in which regions of different characteristics

are to be processed. The computing time, including interpolation of nonprocessed points, was ~ 5 min. on a PC 486 66-MHz microcomputer.

We thank Hans Steinbichler for the images of Fig. 3. This work was partially supported by (Tecnologías Avanzadas de la Producción under project TAP92-0087 from the Comisión Interministerial de Ciencia y Tecnología).

References

1. H. A. Vrooman and A. A. M. Mass, "New image processing algorithms for the analysis of speckle interference patterns," in *Fringe Pattern Analysis*, G. T. Reid, ed., Proc. Soc. Photo-Opt. Instrum. Eng. **1163**, 51–61 (1989).
2. D. Dirksen, X. Su, D. Vukicevic, and G. von Bally, "Optimized phase shifting and use of phase modulation function for high-resolution phase evaluation," in *Fringe '93, Proceedings of Second International Workshop on Automatic Processing of Fringe Patterns*, W. Jüptner and W. Osten, eds. (Akademie, Berlin, 1993), p. 148–153.
3. J. A. Quiroga and E. Bernabeu, "Phase-unwrapping algorithm for noisy phase map processing," *Appl. Opt.* **33**, 6725–6731 (1994).
4. D. P. Towers, T. R. Judge, and P. J. Bryanston-Cross, "A quasi-heterodyne holographic technique and automatic algorithms for phase unwrapping," in *Fringe Pattern Analysis*, G. T. Reid, ed., Proc. Soc. Photo-Opt. Instrum. Eng. **1163**, 95–119 (1989).
5. D. J. Bone, "Fourier fringe analysis: the two-dimensional phase-unwrapping problem," *Appl. Opt.* **30**, 3627–3632 (1991).