

# Un nuovo algoritmo di Phase Unwrapping basato sulla crescita competitiva di regioni

A. Baldi<sup>1</sup>, F. Bertolino<sup>1</sup>, F. Ginesu<sup>1</sup>

<sup>1</sup> Dipartimento di Ingegneria Meccanica – Università di Cagliari  
Piazza d'Armi – 09123 Cagliari  
e-mail: baldi@iris.unica.it

**Keywords:** Phase Unwrapping, Region Growing, Phase Analysis

## Sommario

Il problema del Phase Unwrapping si presenta comunemente quando si ha a che fare con dati sperimentali acquisiti sfruttando fenomeni periodici, a partire dal caso delle tecniche ottico-interferometriche, fino a quelle geologiche e a quelle di analisi in campo medico. Esso è dovuto all'impossibilità di risalire direttamente alla fase del segnale a causa della presenza di operazioni trigonometriche inverse che quindi forniscono i soli valori principali.

Il presente lavoro illustra un nuovo algoritmo per la soluzione di problemi di phase unwrapping fondato sul principio della crescita competitiva di aree limitrofe: partendo da una condizione di equipotenzialità, esso agisce in modo che le varie zone di un campo di fase si accrescano, a scapito di quelle vicine, sulla base di un parametro di qualità che tiene conto dell'estensione e della congruenza di ogni regione.

Dopo una breve descrizione del problema e delle tecniche più note, l'algoritmo viene descritto in dettaglio puntualizzando i principi di base ed alcuni aspetti implementativi che consentono di rendere il metodo competitivo in termini di prestazioni rispetto ai migliori algoritmi noti in letteratura. Il lavoro termina con alcuni esempi di applicazioni su immagini sintetiche.

## Abstract

The phase unwrapping problem arises every time one has to analyse experimental data acquired by means of techniques based on periodical phenomena, from the opto-interferometric ones to some geological or medical techniques.

This paper presents a new phase unwrapping algorithm based on the competitive region growing: starting from an equipotential condition, it makes an area to absorb its neighbours by mean of a quality parameter based on the extension and the coherency of the regions.

After a short description of the phase unwrapping problem and of some of the most used techniques, this paper describes the working principle of the algorithm and the implementation details. The paper ends with some examples of applications of the algorithm on synthetic images.

## 1. INTRODUZIONE

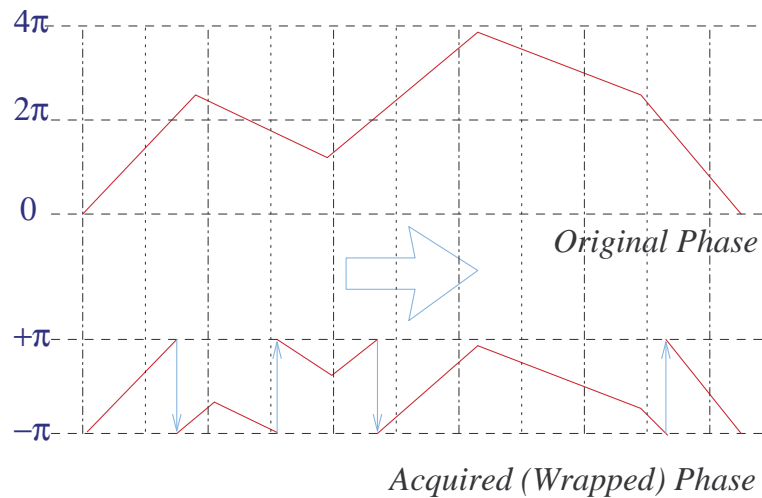
Il problema del Phase Unwrapping si pone ogni qual volta si devono analizzare i dati sperimentali acquisiti con tecniche che sfruttano fenomeni periodici: in questi casi, infatti, l'informazione è di solito contenuta nella fase del segnale, ma questa è nota solo nel suo valore principale, in quanto risultato di operazioni trigonometriche inverse (arcsin, arccos, arctan). Un segnale effettivamente continuo, conterrà quindi una serie di discontinuità nei punti in cui la fase passa da  $2\pi$  a 0 (fig. 1), per cui si parla di *fase modulo  $2\pi$* .

Nel caso monodimensionale, fissato il valore della fase in un punto, l'andamento corretto può essere ricostruito dalla integrazione della fase avendo cura di aggiungere (o sottrarre)  $2\pi$  ogni qual volta si incontra un salto negativo (o positivo) [1]. Tale procedura può essere estesa al caso bidimensionale integrando la fase lungo una linea, per poi passare ad una delle righe limitrofe (linear scanning, multiple scan direction, spiral scanning [2]).

Questo tipo di approccio fallisce in presenza di rumore e/o problemi di aliasing; affinché il problema di phase unwrapping sia ben posto, ossia perché il problema di phase unwrapping abbia un'unica soluzione indipendentemente dal percorso di integrazione (non mettendo in conto la costante iniziale), è necessario che il campo di fase sia irrotazionale [3]. Componenti rotazionali introdotti da rumore e sottocampionamento, infatti, rendono il problema ben più difficile, talvolta insolubile.

Occorre osservare che la natura "vorticoso" dei due tipi di disturbo è essenzialmente differente: disturbi classificabili come "rumore" comportano problemi fortemente localizzati che quindi possono essere compensati sulla base dell'andamento globale del campo; per contro nel caso il segnale sia sottocampionato o ci siano zone

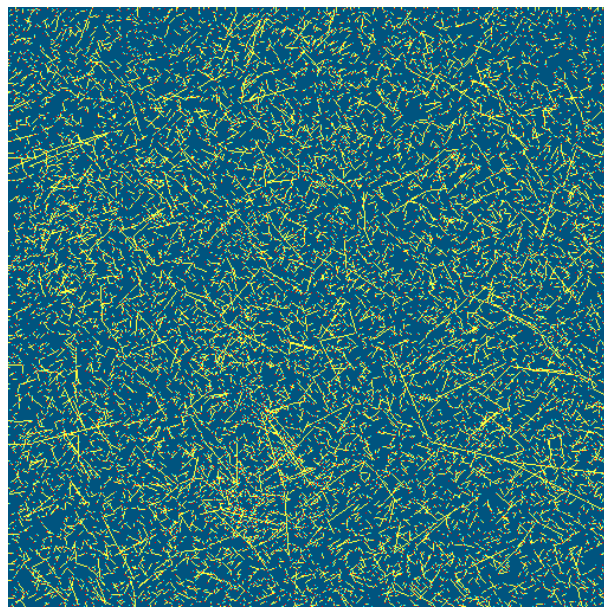
(erroneamente attive) in cui il segnale è assente, l'effetto è globale: in questo caso non esiste un'unica soluzione e diventa difficile dare un giudizio sui risultati ottenuti. Tipicamente si tende a classificare una soluzione come "buona" se essa appare priva di salti, ma è evidente che tale criterio è assolutamente empirico e a volte mendace. In realtà l'unico criterio oggettivo di valutazione delle prestazioni degli algoritmi di unwrapping è il confronto con la soluzione corretta che deve essere nota a priori [4].



**Fig. 1.** Problema del Phase Unwrapping: una funzione continua presenta discontinuità in corrispondenza dei punti in cui la fase passa da  $0$  a  $2\pi$ .

Per quanto riguarda gli approcci al problema, gli algoritmi di Phase Unwrapping possono essere classificati in 5 gruppi a seconda che

1. effettuino una integrazione sequenziale (linear scanning, spiral scanning, quality driven methods [1,5,6]),
2. inseriscano tagli nel dominio in modo da collegare tra di loro sorgenti vorticosi di segno opposto (cut methods [7–10]),
3. trasformino il problema in quello equivalente di Poisson (PCG algorithms,  $L^p$  algorithm [11–13]),
4. dividano il dominio in aree più piccole e semplici, per poi ricomporre la soluzione ottenuta per ognuna di esse (tile method, region method, quad-tree method [1,14,15])
5. usino approcci che non rientrano in nessuna delle categorie precedenti (minimum discontinuity method [16], cellular automata method [17]).



**Fig. 2.** Tagli nel dominio eseguiti dall'algoritmo MLC. Si noti la presenza di segmenti particolarmente lunghi che causano la "segmentazione" del campo di fase.

Gli algoritmi del primo gruppo sono particolarmente soggetti alla propagazione degli errori: poiché l'integrazione è eseguita su base sequenziale (i vari algoritmi differiscono solo per il modo in cui viene scelto il percorso di integrazione), un singolo errore si ripercuote su tutti i punti successivi.

I cut methods introducono tagli nel dominio in modo da renderlo irrotazionale. L'approccio è estremamente interessante—una volta eliminate le componenti vorticose il campo può essere integrato con un qualunque algoritmo sequenziale—ma non esistono evidenze teoriche su quale sia il modo migliore di disporre le linee di separazione: se queste sono mal poste possono isolare completamente delle aree del dominio con conseguenze disastrose (fig. 2).

Gli algoritmi del terzo gruppo minimizzano una funzione di errore in senso globale. Sono generalmente molto affidabili, ma spesso richiedono moli ingenti di calcoli e talvolta necessitano di operazioni di post processing.

Le prestazioni degli algoritmi dei gruppi 4 e 5 sono variabili in quanto diversi sono gli approcci utilizzati: uno dei migliori è il minimum weighted discontinuity che costituisce il termine di confronto per la valutazione comparativa non solo degli algoritmi della stessa famiglia ma anche di tutti gli altri.

L'algoritmo presentato rientra nel quinto gruppo in quanto adotta un approccio per regioni integrato da un criterio di traslazione simile a quello utilizzato dal cellular automata method.

## 2. ALGORITMO DI PHASE UNWRAPPING PER REGIONI

L'algoritmo proposto si basa su due semplici operazioni, eseguite in parallelo da tutte le regioni attive: traslazione ed aggregazione. Data una regione  $i$ , la prima operazione consiste nel sommare alla fase corrente di tutti i punti di  $i$  un multiplo intero di periodi:

$$\forall j / j \in i, \varphi_j' = \varphi_j + 2k\pi \quad (1)$$

con  $k$  numero intero, la seconda nell'accoppiare tra loro due regioni limitrofe, il cui confine sia coerente, in modo da renderle un'unica area.

Partendo da una situazione di massima frammentazione—ogni punto costituisce una regione a se stante—l'applicazione ripetuta delle due operazioni elementari sopra descritte causa la progressiva aggregazione di aree limitrofe, sino a giungere alla creazione di macro regioni che, traslando in senso verticale, si aggregano a loro volta tra loro arrivando alla definizione di un'unica area.

Le operazioni elementari su cui l'algoritmo si basa sono in linea di principio semplici, ma presentano alcuni aspetti da definire:

- l'algoritmo proposto è di tipo parallelo (potrebbe essere facilmente implementato su macchine multiprocessore), ma deve poter essere eseguito in modo sequenziale: è dunque necessario un criterio che specifichi l'ordine in cui le regioni devono diventare attive;
- l'operazione di traslazione necessita di un criterio che specifichi la direzione in cui ci si deve spostare, in altri termini serve una funzione che permetta di determinare il valore di  $k$  nella eq. (1);
- l'operazione di aggregazione richiede un criterio che consenta di stabilire se esiste, e nel caso quale è, la regione limitrofa più adatta ad essere conglobata alla corrente;
- infine è necessario un criterio che consenta di decidere se si deve eseguire una operazione di traslazione od una di aggregazione.

Per quel che riguarda l'ultimo punto, non è possibile stabilire un criterio oggettivo: infatti, a seconda della situazione, è preferibile l'una o l'altra delle operazioni elementari. A questo proposito, per evitare di dover effettuare analisi dispendiose dal punto di vista computazionale, si è deciso di eseguire in successione entrambe le operazioni, prevedendo per ognuna di esse la possibilità di mosse "nulle". In quest'ottica è evidente che l'operazione di traslazione è propedeutica all'altra, in quanto consente di annullare eventuali discontinuità presenti tra la regione corrente e quella limitrofa; si è pertanto deciso di consentire un numero illimitato di traslazioni unitarie seguite della verifica della congruenza dei confini. Le traslazioni proseguono fin tanto che viene richiesto uno spostamento nullo o di segno opposto al precedente. Quest'ultima condizione serve ad evitare il possibile insorgere di cicli infiniti nel caso in cui al passo  $n$  si esegua una traslazione in un verso ed al passo  $n+1$  una nel verso opposto.

Per quel che riguarda l'ordine di attivazione delle aree, si è ritenuto che il criterio più idoneo a simulare un'esecuzione parallela fosse quello di rendere attive le regioni secondo una sequenza casuale. In realtà si è anche investigata la possibilità di scegliere un criterio di merito, ma i vari tentativi fatti (ordinamento sulla base della coerenza interna, delle dimensioni, della lunghezza del confine in rapporto al numero di punti appartenenti all'area) hanno dato risultati essenzialmente equivalenti al semplice ordinamento casuale e comunque dipendenti dallo specifico test.

Per stabilire la direzione di traslazione, si è semplicemente verificato il salto di fase associato alla posizione corrente, a quella corrispondente ad una traslazione unitaria positiva e a quella corrispondente ad una traslazione negativa, per poi muoversi nel verso indicato dal salto inferiore.

Il parametro di merito  $m_l$  associato alle 3 posizioni in esame (corrente, traslata verso l'alto, traslata verso il basso) può essere facilmente stimato con l'ausilio di un vettore  $\mathbf{V} = (0, +1, -1)$  tramite il quale il salto di fase può essere calcolato come

$$m_l = \sum_j \left| \varphi_{k,j} - (\varphi_{i,j} + 2\pi V_l) \right| \quad (2)$$

con  $l = 1..3$ ,  $j$  esteso a tutti i punti posti sul bordo dell'area  $i$  corrente e  $k$  la generica area limitrofa.

Se  $\lambda$  è l'indice del valore inferiore di  $m$ , il risultato del test sarà  $V_\lambda$ .

La parte più critica dell'algoritmo è la fase di aggregazione: tra le  $n$  aree limitrofe devono essere selezionate quelle congruenti, ovvero quelle che si trovano a distanza inferiore ad un periodo (I) e di queste ultime deve essere scelta la migliore cioè l'area che presenta la minima discontinuità media (II).

Sia la prima che la seconda parte dell'operazione si prestano ad errori in presenza di rumore: poiché infatti in questo caso è comune che esistano punti non allineati al resto dell'area, è innanzitutto necessario che il controllo di congruenza sia effettuato in senso medio e, in secondo luogo, occorre stabilire come debba essere calcolato il criterio di merito. A questo proposito, tra le varie soluzioni possibili (sommatoria della differenza di fase, calcolo del numero di salti, funzioni quadratiche), quella più efficace è risultata l'utilizzazione di un criterio misto che tiene conto sia della estensione del perimetro comune tra le aree in esame che del salto di fase. La funzione di merito utilizzata è:

$$a = \sum_j \pi - \left| \varphi_{k,j} - \varphi_{i,j} \right| \quad (3)$$

con  $i$  area corrente,  $k$  area candidata e  $j$  estesa a tutti i punti comuni tra  $i$  e  $k$ .

Se il salto di fase tra due punti è maggiore di  $\pi$ , la funzione all'interno della sommatoria risulta negativa, mentre è pari a  $\pi$  in caso di salto nullo. Poiché  $a$  è massima in caso di aree estese con salti ridotti, il controllo di ammissibilità si riduce a verificare che essa sia positiva, mentre quello di merito deve semplicemente scegliere la regione che fornisce il valore più grande.

In definitiva l'algoritmo consiste nell'esecuzione iterativa dei seguenti passi:

1. Ciclo sulle aree: tra tutte le aree attive che non sono state ancora selezionate, se ne sceglie una a caso.
2. Si seleziona la direzione di traslazione per l'area corrente.
3. Se la direzione di traslazione non è congruente alla precedente o nulla si va al passo 5.
4. Si esegue la traslazione e si torna al passo 2.
5. Tra le aree limitrofe a quella corrente si sceglie l'area da aggregare.
6. Si accorpa l'area selezionata e si torna al passo 1.

Il ciclo si interrompe quando durante l'esecuzione dei passi 1-6 non è stata eseguita neanche un'operazione di traslazione o di aggregazione (o, ovviamente, se rimane solo un'area).

## 2.1 Pesi

Le equazioni 2 e 3 possono essere facilmente modificate introducendo una funzione che "pesi" i salti di fase sulla base dell'affidabilità dell'area in cui ci si trova. In tal modo le prestazioni dell'algoritmo risultano molto aumentate e si ha il vantaggio accessorio che è possibile disabilitare zone per le quali si sa a priori che non esiste informazione o questa è talmente disturbata da non poter essere utilizzata.

La versione "pesata" dell'eq. 2 è ricavabile scalando il salto di fase per il peso corrispondente:

$$m_l = \sum_j w_{k,j} w_{i,j} \left| \varphi_{k,j} - (\varphi_{i,j} + 2\pi V_l) \right| \quad (2')$$

dove  $w_{k,j}$  e  $w_{i,j}$  sono i pesi associati a due generici pixel appartenenti l'uno all'area  $k$  e l'altro all'area  $i$ .

La versione modificata della eq. 3 è apparentemente simile:

$$a = \sum_j w_{k,j} w_{i,j} \left( \pi - \left| \varphi_{k,j} - \varphi_{i,j} \right| \right) \quad (3')$$

ma corrisponde ad un concetto diverso: in questo caso non viene pesato il salto di fase, ma l'effetto che questo ha nella determinazione del parametro globale di merito. Se si fosse pesato direttamente il  $\Delta\varphi$ , infatti, ad un basso peso sarebbe stato associato (erroneamente) un alto valore del parametro indipendentemente dal salto di fase effettivamente presente.

### 3. DETTAGLI IMPLEMENTATIVI

Una efficiente implementazione dell'algoritmo esposto nel paragrafo precedente richiede:

- che sia possibile determinare in modo semplice a quale regione appartiene un pixel;
- che sia possibile individuare i pixel di una data area;
- che sia possibile stabilire se un pixel è interno o sul bordo di un'area (un pixel appartiene al bordo di una regione se confina con pixel che non appartengono alla regione stessa. Si noti che in base a questa definizione le regioni non sono semplicemente connesse).

Mentre la prima richiesta può essere soddisfatta utilizzando una matrice  $\mathbf{A}$ , i cui elementi  $A[i]$  indichino a quale area deve essere associato il punto in esame (si noti che si è utilizzato una rappresentazione mono-dimensionale della matrice in cui l'elemento  $i,j$  è posto nell'elemento  $i \cdot n_c + j$ , con  $n_c$  numero di colonne), le richieste successive pongono problemi maggiori: in linea di principio l'utilizzo della matrice  $\mathbf{A}$  è sufficiente a determinare i pixel di un'area, ma per far ciò è necessario attraversarla tutta controllando uno ad uno tutti gli elementi. Ancor più difficoltosa è la determinazione dei "confini", in quanto non solo si dovrebbe controllare a che area è associato ciascun pixel, ma, una volta appurata l'appartenenza alla regione di interesse, occorrerebbe verificare che almeno uno dei pixel limitrofi sia associato ad una regione diversa da quella in esame.

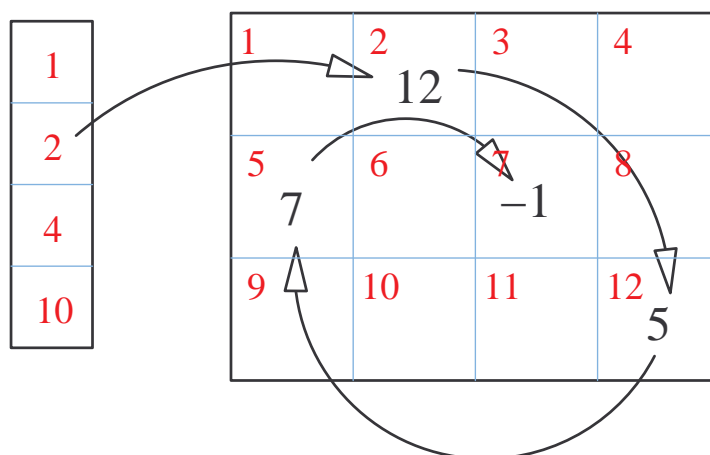


Fig. 3. Struttura della matrice di adiacenza.

Per rendere più efficiente l'esecuzione delle operazioni di identificazione e selezione dei pixel, si è quindi fatto ricorso ad una matrice ausiliaria  $\mathbf{B}$  strutturata come illustrato in fig. 3: ogni regione è associata ad un indice  $l$  che identifica un punto interno alla regione stessa; in  $B[l]$  è posto l'indice del punto successivo e così via. Si crea in tal modo una catena che presenta come ultimo termine un valore negativo, i cui membri sono tutti e solo punti della medesima regione. In tal modo, per aggregare due regioni, è sufficiente modificare l'indice dell'ultimo elemento della prima catena.

Occorre osservare che essendo fisso il numero di punti del dominio, le dimensioni della matrice  $\mathbf{B}$  non possono variare: i suoi elementi passano da una regione ad un'altra cosicché non è necessaria alcuna operazione di allocazione della memoria.

In modo del tutto analogo possono essere gestite le aree di confine delle regioni, anche se in questo caso l'operazione di aggregazione presenta elementi di maggiore complessità: i punti che durante tale operazione diventano interni, infatti, devono essere rimossi.

Per quanto riguarda la selezione casuale delle regioni, prevista al punto 1 dell'algoritmo, occorre osservare che il semplice utilizzo di un generatore di numeri casuali è inefficiente se si vuole che tutte le regioni attive vengano estratte una volta per ciclo. L'impiego di un vettore indicante lo stato (estratto/non estratto) delle singole regioni, in effetti, consente di marcare tutte quelle già attivate, ma tale approccio con l'aumentare delle aree selezionate, richiede un numero sempre crescente di estrazioni di candidati, rallentando in modo inaccettabile l'esecuzione dell'algoritmo. Per evitare questo inconveniente occorre affiancare alla lista delle regioni attive un intero  $n$  indicante il numero di quelle che devono essere ancora estratte. Ogni volta che viene selezionata un'area,  $n$  viene decrementato e l'area stessa viene scambiata con l'ultima attiva. In tal modo i primi  $n$  elementi sono tutti e soli quelli che devono ancora essere estratti ed è possibile selezionarli facilmente:

$$i = p \% n \quad (4)$$

dove  $p$  è un numero casuale e il  $\%$  indica il resto della divisione intera.

Si noti che durante l'operazione di aggregazione, un'area, potenzialmente ancora da selezionare, viene a scomparire. E' quindi necessario aggiornare contestualmente l'elenco delle regioni attive ed eventualmente



decrementare  $n$ . Poiché l'operazione di scambio rende in breve tempo il vettore delle aree attive non ordinato, per individuare l'elemento da rimuovere è necessario attraversarlo completamente. Alternativamente è possibile introdurre un vettore contenente la funzione inversa del vettore delle aree attive che mappa le aree nell'indice del vettore; in tal modo la ricerca lineare può essere evitata al prezzo della gestione di questa nuova struttura dati.

#### 4. ESEMPI DI APPLICAZIONE

Nelle figure 4–9 sono mostrati vari stadi di esecuzione dell'algoritmo. In particolare in fig. 4 è rappresentata una ipotetica fase modulo  $2\pi$  generata a partire di una funzione sinc [18]. Sono stati artificialmente introdotti disturbi sia sulla modulazione che sul valor medio; inoltre si è aggiunto rumore bianco in modo da alterare due bit del segnale (campionato ad 8 bit). Tali disturbi sono distribuiti su tutta l'immagine e sono particolarmente evidenti nella parte sinistra della stessa.

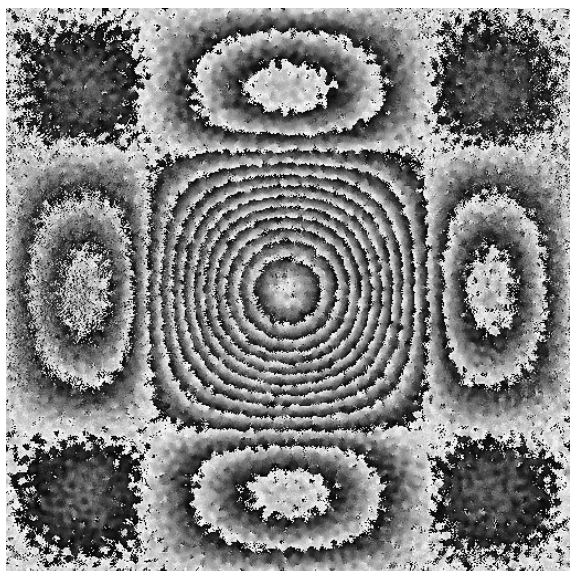


Fig. 4. Fase modulo  $2\pi$

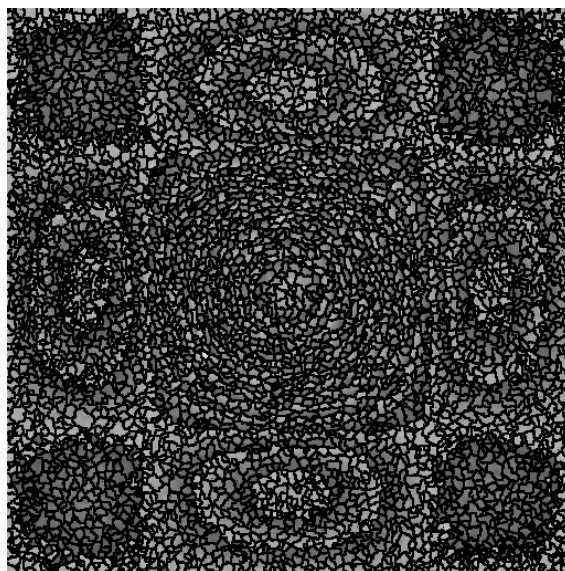


Fig. 5. Passo 2

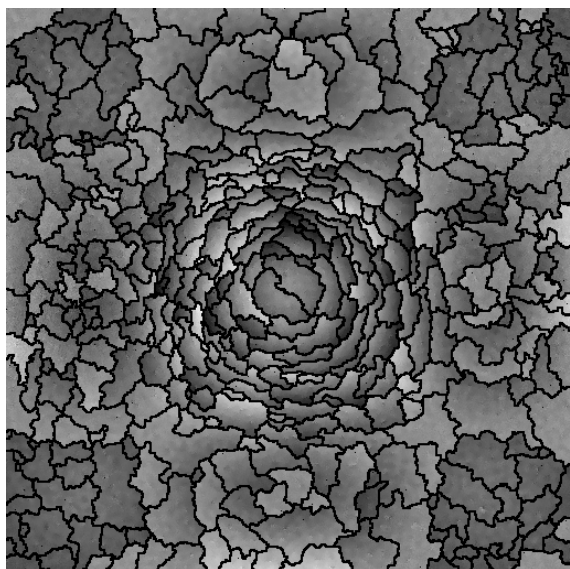


Fig. 6. Passo 5

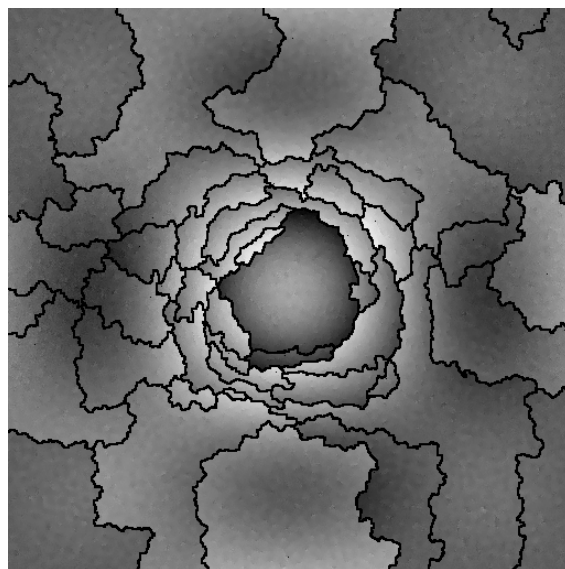


Fig. 7. Passo 8

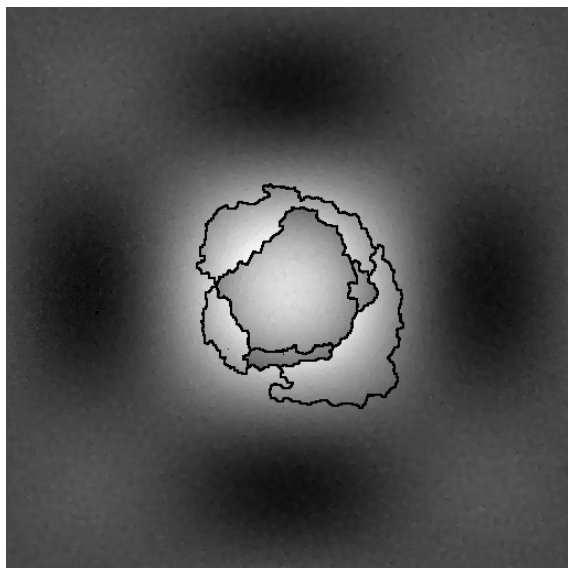


Fig. 8 Passo 12.

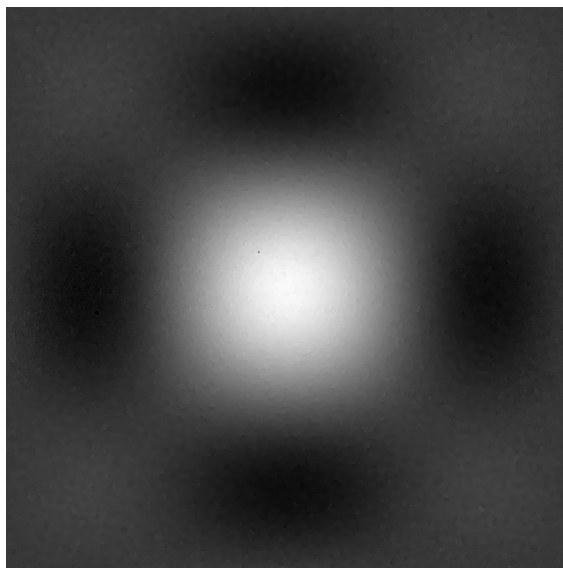


Fig. 9. Passo 19.

Le figure 5–9 corrispondono a differenti stadi di avanzamento dell’algoritmo: la fig. 5 è un’istantanea della distribuzione delle regioni alla fine della seconda iterazione; le altre corrispondono rispettivamente alla fine della quinta, ottava, dodicesima e diciannovesima iterazione. Se in fig. 5 sono ancora attive varie migliaia di aree (il campo di fase iniziale è di  $512 \times 512$  pixel, corrispondente a circa 256000 aree inizialmente attive), già in corrispondenza dell’ottava iterazione (fig. 7) queste si sono ridotte a poche decine per giungere ad una sola alla fine dell’esecuzione (iterazione 19, fig. 9).

In figura 7 sono particolarmente evidenti le varie macro aree: esse tendono a svilupparsi a quote diverse le une dalle altre e solo quando non sono più possibili ulteriori accrescimenti esse traslano verticalmente, in modo da rendere congruenti i propri confini con quelli delle aree limitrofe (I diversi toni di grigio corrispondono a diverse altezze: nero per la  $z$  minima, bianco per la massima). Ad illustrare questa particolarità si prenda in considerazione la macro area corrispondente al vertice della sinc: essa si trova ad una quota molto bassa in quanto l’alto numero di frange tende ad isolarla dal resto del campo di fase. Perché trasli nella sua posizione definitiva occorre aspettare l’ultimo passo (fig. 9). Questo tipo di comportamento è caratteristico di aree quasi isolate: una macro area infatti è tanto più persistente (nel senso che non viene aggregata ad altre) quanto più disturbato è il confine che la delimita.

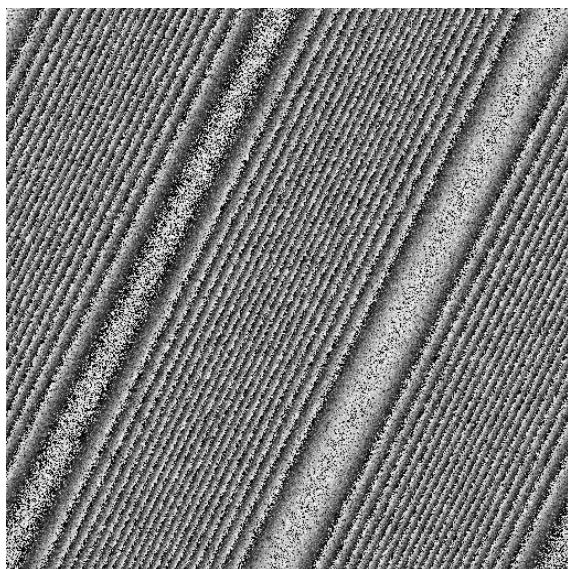
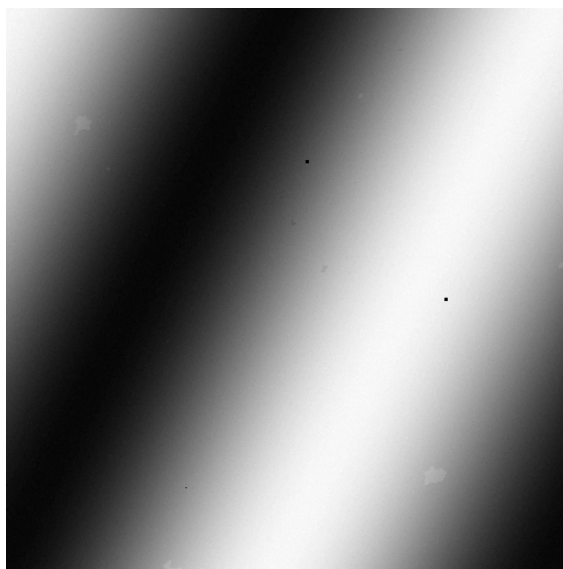
Fig. 10. Fase modulo  $2\pi$ 

Fig. 11. Soluzione dopo 31 iterazioni

In figura 10 ed 11 è illustrata l’applicazione dell’algoritmo ad un altro test. Malgrado il livello di rumore relativamente basso, il campo di fase di fig. 10 presenta difficoltà non trascurabili per l’alto numero di frange presenti sui fianchi della sinusoida. Anche in questo caso l’algoritmo, a parte due piccoli errori, ha fornito la soluzione corretta. Si noti però che per giungere alla convergenza sono state necessarie 31 iterazioni.



## 5. CONCLUSIONI

In questo lavoro è stato presentato un nuovo algoritmo di phase unwrapping basato sulla crescita competitiva di regioni. Il campo di fase, inizialmente suddiviso in un numero elevatissimo di regioni (una per ogni punto) viene progressivamente riassembleto in aree di dimensioni sempre maggiori derivanti dall'aggregazioni di regioni più piccole. Oltre ad aggregarsi, le varie aree traslano per rendere congruenti i propri confini con quelli delle regioni limitrofe. Per questa operazione si sfrutta un algoritmo simile a quello utilizzato nel cellular automata method ma, a differenza di quest'ultimo, non sono possibili oscillazione e l'algoritmo giunge sempre in convergenza.

I tempi di esecuzione dipendono fortemente dalla struttura del campo di fase (oltre che, ovviamente, dalle dimensioni dello stesso) ma, a prezzo di una qualche complessità a livello implementativo, sono competitivi rispetto ad algoritmi di pari prestazioni.

Dal punto di vista dell'accuratezza della soluzione, il metodo proposto è estremamente robusto ed è in grado di fornire risultati comparabili (o migliori) di quelli degli algoritmi di riferimento, con tempi di esecuzione di meno di un terzo di questi ultimi. Solo in presenza di densità di frange estremamente elevate è possibile che l'algoritmo proposto vada in crisi. In questi casi esso si arresta quando sono attive ancora due o più aree per cui tale condizione è estremamente semplice da individuare.

Resta da analizzare l'effetto dei differenti pesi sulle prestazioni dell'algoritmo. In questo campo sono al momento in corso studi comparativi.

## Bibliografia

- [1] K. Itho: Analysis of the phase unwrapping problem. *Appl. Opt.* **21** (14), 2470 (1982)
- [2] D.W. Robinson: "Phase unwrapping methods", in: *Interferogram analysis: Digital Fringe Pattern Measurement Techniques*, D.W. Robinson and G.T. Reid eds, Institute of Physics Publishing, Bristol and Philadelphia, 194–229, 1993
- [3] D.C. Ghiglia and M.D. Pritt, *Two-Dimensional Phase Unwrapping. Theory, Algorithms, and Software*, Wiley, New York, 1998.
- [4] A. Baldi, F. Bertolino and F. Ginesu, "On the Performance of some Unwrapping Algorithms", accepted for publication on *Optics and Lasers in Engineering*.
- [5] N.H. Ching, D. Rosenfeld and M. Braun, "Two-dimensional phase unwrapping using a minimum spanning tree algorithm", *IEEE Trans. Image Processing*, **1**, 355–365 (1992)
- [6] M. Takeda and T. Abe: Phase unwrapping based on maximum cross-amplitude spanning tree algorithm: a comparative study. In: *Interferometry VII: Techniques and Analysis*, M. Kujawinska, R.J. Pryputniwicz and M. Takeda eds, SPIE Proc. **2544**, 122–129, 1996.
- [7] R.M. Goldstein, H.A. Zebker and C.L. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping", *Radio Science* **23** (4), 713–720 (1988)
- [8] J.M. Huntley, "Noise immune phase unwrapping algorithm", *Appl. Opt.* **28** (15), 3268–3270, (1989).
- [9] R. Cusack, J.M. Huntley and H.T. Goldrein, "Improved noise-immune phase unwrapping algorithm", *Appl. Opt.* **28**, 781–789 (1995).
- [10] J.R. Buckland, J.M. Huntley and S.R.E. Turner, "Unwrapping noisy phase maps by use of a minimum cost matching algorithm.", *Appl. Opt.* **34** (23), 5100–5108 (1995).
- [11] D.C. Ghiglia and L.A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods", *J. Opt. Soc. Am. A* **11**, 107–117 (1994).
- [12] M.D. Pritt, "Phase unwrapping by means of multigrid techniques for interferometric SAR", *IEEE Trans. Geosci. Remote Sens.* **34** (3), 728–738 (1996).
- [13] D.C. Ghiglia and L.A. Romero, "Minimum Lp-norm two dimensional phase unwrapping", *J. Opt. Soc. Am. A* **13** (10), 1–15 (1996).
- [14] J.J. Gierloff, "Phase unwrapping by regions", in *Current Development in Optical Engineering II*, R.E. Fischer and W.J. Smith. Eds, SPIE Proc. **818**, 2–9, 1987.
- [15] A. Baldi, "Two-dimensional phase unwrapping by quad-tree decomposition", *App. Opt.*, **40**, (8) 1187–1194 (2001).
- [16] T.J. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity", *J. Opt. Soc. Am.* **14** (10), 2692–2701 (1997).
- [17] D.C. Ghiglia, G.A. Mastin and L.A. Romero, "Cellular-automata method for phase unwrapping", *J. Opt. Soc. Am. A* **4** (1), 267–280 (1987).
- [18] A.R. Weeks and H.R. Myler, "Computer-generated noise images for the evaluation of image processing algorithms", *Opt. Eng.* **32** (5), 982–992 (1993)