PHASE UNWRAPPING ALGORITHMS IMPLEMENTED ON A SYMBIAN BASED MOBILE DEVICE

Carol RUS, Lacrimioara GRAMA, Andrei DUSLEAG, Corneliu RUSU
Technical University of Cluj-Napoca, Romania
Email: Lacrimioara.Grama@bel.utlcuj.ro

Abstract: Phase unwrapping is a procedure that allows 3D representation for surfaces when the data is acquired using interferometry based techniques. An interferogram contains phase difference values wrapped onto a fixed range of angles 0-360 degrees. In order to compute surface heights and generate an elevation model, the interferogram have to be unwrapped. This paper presents two 2D phase unwrapping algorithms: the zig-zag approach and the least-squares method, and shows an implementation of them on a mobile device based on Symbian S60 platform. Also, filtering methods (i.e mean filter and vector filter) for preprocessing the interferogram are discussed and implemented. The implementation was tested using SAR (Synthetic Aperture Radar) interferograms. Based on the results, it can be concluded that mobile devices offer enough processing and graphical power and development options, in order to be a good choice to test some new signal processing algorithms.

Key words: *interferogram, DEM, phase unwrapping, Symbian, S60.*

I. INTRODUCTION

Interferometry is the technique of superimposing (interfering) two or more waves, to detect differences between them. Interferometry is applied in a wide variety of fields, including astronomy, fiber optics, optical metrology, oceanography, seismology, quantum mechanics and plasma physics [1].

Interferometry works because two waves with the same frequency that have the same phase will add to each other while two waves that have opposite phase will subtract. Typically, in an interferometer, a wave is split into two (or more) coherent parts, which travel different paths, and the parts are then combined to create interference.

When the paths differ by an even number of half-wavelengths, the superposed waves are in phase and interfere constructively, increasing the amplitude of the output wave.

When they differ by an odd number of half-wavelengths, the combined waves are 180 degrees out of phase and interfere destructively, decreasing the amplitude of the output.

Thus anything that changes the phase of one of the beams by only 180 degrees shifts the interference from a maximum to a minimum. This makes interferometers sensitive measuring instruments for anything that changes the phase of a wave, such as path length or refractive index [2].

Radar interferometry (InSAR) has been successfully applied to measure the surface topography in the past two decades [3]. Since InSAR is a coherent technology, the most important constraint for the radar echo is coherence. However, decorrelation is always inevitably introduced by the thermal noise, geometric decorrelation, Doppler centroid decorrelation, volume scattering decorrelation, temporal decorrelation, orbit error, atmospheric delay, and data processing noise. The decorrelation in the interferogram then has a significant effect on the accuracy of InSAR derived digital elevation model (DEM).

Noise filtering has been commonly applied to the interferogram either in frequency domain or in spatial domain, but some noise will still remain in the DEM products [4]. So low-pass filtering, e.g. median filtering is also necessary for the InSAR DEM. In company with the removal of noise, median filter also tends to smooth out the details in the image.

This work focuses on adapting and implementing two phase unwrapping approaches for a Symbian S60 mobile device: a simple one – the zigzag algorithm, and a more complex one – the least squares method. Also, the preprocessing step (i.e. noise filtering) is implemented on the mobile device.

In our days, mobile devices come with more and more computing power. Some of them (e.g. Symbian S60 based devices) offer good facilities for research purposes by allowing the user to add own implementations to run on the device. Unfortunately, there is a lack of research papers discussing signal processing algorithms implemented on Symbian publicly available. This happens mainly due the fact that Symbian is a rather new option for researchers, but also due to political (and commercial) reasons since

Manuscript received March 13, 2008; revised June 27, 2008

Symbian is sustained by a group of companies and it is not a 100% open system like Linux. The most reliable source of documentation for S60 platform is Forum Nokia [9] since it is developed by Nokia upon Symbian OS. There, one can find good articles upon 3D graphics in S60 and image handling in S60, topics used by our implementation.

Signal processing algorithms implemented on S60 platform are able to run fast, at speeds comparable to Matlab or even faster, if we have the right API(s) and if the amount of data does not overtake phone's resources. The S60 APIs offer less complex access to the phone resources (the most important for our case are the DSPs). The mobile device's resources are large enough to handle large images and videos. Also, the display capabilities provided by S60 phones are good. For our problem, the necessary APIs were publicly available at [9].

The paper is organized as follows. In Section II we talk about interferogram filtering methods (mean, median and vector filtering), and then we will present the phase unwrap metods for interferograms. In Section III we will discuss obout the zigzag method, and in Section IV about the least-squares difference criteria. In Section V the Symbian S60 platform is presented, and in Section VI the experimental results are shown. In the end of the paper we will drag conclusions (Section VII).

II. INTERFEROGRAM FILTERING

The mean filter is a simple sliding-window spatial filter that replaces the center value in the window with the average (mean) of all the pixel values in the window. The window, or kernel, is usually square but can be any shape.

The median filter is also a sliding-window spatial filter, but it replaces the center value in the window with the median of all the pixel values in the window. As for the mean filter, the kernel is usually square but can be any shape.

The advantage of mean and median filters is their simplicity. However, the main disadvantage is that they do not adapt the local noise level variations and the geometrical shapes of the objects within the image.

Vector filtering is one of the best methods for interferogram filtering, because it does not affect the image energy, being rapidly and efficiently [2].

Vector filtering is defined as:

$$e^{j\phi} = \frac{1}{mn}\sum_{k=1}^{m}\sum_{l=1}^{n} e^{j\phi_{k,l}} \quad (1)$$

where ϕ is the filtered phase, $\phi_{k,l}$ is the pixel's (k, l) phase in the filtering window.

The first step for filtering is to map the phase $\phi_{k,l}$ on a vector $e^{j\phi_{k,l}}$, that can be write as

$$e^{j\phi_{k,l}} = \cos\left(\phi_{k,l}\right) + j\sin\left(\phi_{k,l}\right) \quad (2)$$

The interferogram is mapped on a sinusoidal, respectively cosinusoidal image. After that, the images are filtered using a median filtering method with a window of same dimension [2].

The last step consists in the filtered phase evaluation according to next relation

$$\phi = \arctan\left(b/a\right) \quad (3)$$

where b represents a pixel from the sine image, and a represents a pixel from the cosine image.

III. ZIGZAG PHASE UNWRAPPING

Phase unwrapping is the key procedure for SAR interferometry data evaluation with respect to establishing DEM's. Meanwhile, there are many methods for the 2D phase unwrapping. All of the commonly used phase unwrapping algorithms relate the phase values by first differentiating the phase field and subsequently reintegrating, adding back the missing integral cycles to obtain a more continuous result [3].

The many algorithms proposed for phase unwrapping over the past few years fall into two basic classes: algorithms based on following the paths, and algorithms that derive a smooth field by integrating the gradient of the observations subject to smoothness constraints as determined by least squared difference criteria. The phase signal in the interferogram is given modulo 2π [4]. Thus, the correct integer number of the whole phase cycles has to be found to obtain the absolute phase signal.

For DEM without residua, the phase unwrap can be done by a path following algorithm, in our case a zigzag one. The input parameter is the interferogram's phase, and to the output we will obtain the unwrap phase.

We consider a MxN interferogram and we follow it in zigzag, and then summing the phase differences. We can unwrap the phase for this interferogram if we start following the path with the first row and used as intial condition the last value from that row to unwrap the phase on the next row, following it in the inverse direction. The zigzag algorithm path is presented in Figure 1.

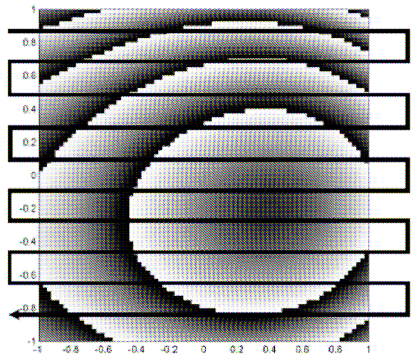

Figure 1. The zigzag algorithm path

For the first pixel from the image we have to chose an initial value. This value can be arbitrarily choosen, because we are interested by the phase differences, not in the absolute value.

IV. LEAST-SQUARES PHASE UNWRAPPING

A more sophisticated principle is the least-square method which minimizes the differences between the derivatives of the wrapped and unwrapped phase functions using the fact that the minimum of the first derivative is determined by the null of the second derivative. In the 2D case the partial differential quotients have to be used. This method seems to be very simple and fast. However, the errors are propagating very strongly down to the entire sequence and therefore, this method is not very practicable for large area processing.

The least-square phase unwrapping algorithm was presented by Ghiglia and Romero [5], who applied a mathematical formalism first developed by Hunt [6] to the radar interferometry phase unwrapping problem. Hunt developed a matrix formulation suitable for general phase reconstruction problems; Ghiglia and Romero found that a Discrete Cosine Transform (DCT) technique permits accurate and efficient least-squares inversion, even for the very large matrices encountered in the radar interferometry special case. They examined both unweighted and weighted least-squares solution procedures.

In the least squares methods, the vector gradient of the phase field is determined and then integrated, subject to the constraint of a smooth solution. Any value may be added to ensure smoothness and continuity in the solution. Thus the spatial error distribution may differ between the approaches, and the relative merits of each method must be determined, depending on the application.

The unweighted algorithm is implemented as follows. Consider a sampled wrapped phase function $\varphi_{k,l}$, evaluated at discrete points (k, l) corresponding to the row and column locations, respectively, of a 2D data matrix.

We want to determine a smooth, unwrapped phase function $\phi_{k,l}$ that minimizes the difference between the gradients calculated from the wrapped phase and the presumed smooth, unwrapped phase. Hunt [6] shows that these may be related by a matrix-vector equation

$$\mathbf{s} = \mathbf{P}\phi + \mathbf{n} \tag{4}$$

in which **s** is derived from the measured row and column phase differences of φ, **P** is a matrix containing 1's, -1's, and 0's describing row and column differencing operations, ϕ is the unwrapped phase field, and **n** is a vector representing measurement noise. The least-squares solution is the well-known

$$\phi = \left(\mathbf{P}^T \mathbf{P}\right)^{-1} \mathbf{P}^T \mathbf{s} \tag{5}$$

Specifically, in the radar interferometry case we minimize the function

$$\begin{gathered}\sum_{k=0}^{M-2}\sum_{l=0}^{N-1}\left(\phi_{k+1,l} - \phi_{k,l} - \Delta_{k,l}^{x}\right)^2 + \\ +\sum_{k=0}^{M-1}\sum_{l=0}^{N-2}\left(\phi_{k,l+1} - \phi_{k,l} - \Delta_{k,l}^{y}\right)^2\end{gathered} \tag{6}$$

where $\Delta_{k,l}^{x}$ and $\Delta_{k,l}^{y}$ are the row differences and column differences of the wrapped phases, respectively.

The least-squared error solution is obtained by differentiating Eq. (6) with respect to $\phi_{k,l}$ and setting the result equal to zero

$$\begin{gathered}\left(\phi_{k+1,l} - 2\phi_{k,l} + \phi_{k-1,l}\right) + \left(\phi_{k,l+1} - 2\phi_{k,l} + \phi_{k,l-1}\right) = \\ = \left(\Delta_{k,l}^{x} - \Delta_{k-1,l}^{x}\right) + \left(\Delta_{k,l}^{y} - \Delta_{k,l-1}^{y}\right)\end{gathered} \tag{7}$$

Eq. (7), the unweighted case, is a discrete form of Poisson's equation and may be solved efficiently by using a DCT approach [4].

Now, if some points in an interferogram are deemed more reliable than others, for example, possessing higher correlation, a weighted algorithm may be used. Including in Eq. (4) a matrix **W** of weights for each point yields

$$\mathbf{Ws} = \mathbf{WP}\phi + \mathbf{n} \tag{8}$$

We obtain

$$\phi = \left(\mathbf{P}^T \mathbf{W}^T \mathbf{WP}\right)^{-1} \mathbf{P}^T \mathbf{W}^T \mathbf{Ws} \tag{9}$$

Unfortunately, the weighted least-squares equation does not reduce to the same simple Poisson's equation form, and thus the efficient DCT algorithm cannot be directly applied. Ghiglia and Romero [5] show, however, that several iterative approaches using repeated DCTs are yet relatively efficient at achieving an accurate solution.

The accuracy of the solution depends on the choice of weights to apply to each point of the measured phase. Choosing the same value for all weights reduces the problem to the unweighted solution. More typical choices are functions of the signal-to-noise ratio or the observed interferometric correlation, both of which give greater emphasis to the phase estimates deemed more reliable. Various methods for selecting weights are described by Pritt [7].

All of the least-squares algorithms produce a continuous solution unless zero weights are assumed at some data points. In fact this constraint is fundamental to the way the problem is framed. Thus we would expect the algorithm to perform poorly in the presence of actual discontinuities in the underlying phase field, as can be present if there is any layover or extreme foreshortening in the radar image, unless the locations of the zero weights are properly and carefully chosen.

Weighted solutions derived from correlation or signal-to-noise measures can lessen the errors by tying down the solution less in the discontinuous areas; but some, albeit smaller, smoothness is assumed everywhere the weights are nonzero [4]. If the continuity constraint is removed completely in laid over regions but not in the remainder of the image, the weighted least-squares solution can minimally distort the result.

V. SYMBIAN S60 PLATFORM

Symbian is an operating system for mobile devices written in C++. Symbian was designed so that it can fulfill some important features like optimized runtime memory usage, battery life, many communication capabilities, multitasking based on less computing power than in case of a desktop PC, many communication capabilities, and support for internationalization. Based on Symbian, different mobile devices manufacturers produced their own platforms to integrate with their own devices – e.g. Series 60 (shortly S60) platform popular in Nokia devices, and UIQ platform running on Sony Ericsson devices.

Our work was concerned with S60 platform. S60 devices (and in general Symbian based devices) offer the possibility of development and installation of new applications so they are a good choice for research activities.

For the S60 platform, a public software development kit (SDK) is offered at no cost. It contains libraries, command line tools, and emulator [8, 9]. It runs on MS Windows and Perl. An Integrated Development Environment (IDE) is also available freely: Carbide Express. The developer can add extra libraries to the environment. Applications are deployed (installed) on the phone using Bluetooth interface (or cable) and driver software from the manufacturer.

In our work, we used C++ (Symbian C++) – this is the language that can get the maximum from Symbian/S60 OS. The platform can run also applications developed in other languages as Java, Perl, standard C, which may be easier programming solutions, but they cannot access all phone's resources [8].

Symbian C++ uses naming conventions for more readability of the code and they are reflected in types' names, class names, and, optionally, in variable names. Phones are devices which are restarted rarely and they include a limited amount of memory. For this reason, the resource allocation and deallocation is handled using own Symbian style for object construction and exception handling. Text is cared in an efficient manner using descriptors.

A Symbian application does not use multiple threads because they can cause the phone to stop responding to user commands for awhile. This issue is solved using an asynchronous solution the so called active objects. Client/Server architecture is a model used by Symbian to split the UI (User Interface) from the engine, but it is, also, a technique used to provide different APIs (Application Programming Interfaces) to the developers.

The first versions of the S60 platform are restrictions free in accessing the phone's resources. But starting with the 3rd version of the platform, access to APIs and system resources is done using capability certificates to sign the application before installation on the phone [9]. Basic capabilities like accessing communication interface, reading and writing user files, accessing graphical resources, implementing some algorithms are given through so called self-signing process that is free.

When choosing an S60 SDK version, the type of the specific target device must be considered (e.g. N70 phone likes applications built with S60 SDK 2nd FP3, N80 phone will run applications developed with S60 SDK 3rd MR). The SDK offers many APIs that allow the developer to use all the phones DSPs and memory in an efficient manner. An S60 API includes header files (text files with definition of classes, functions and other types), and lib files (binaries described by the headers) that come in two versions: for emulator and for the device.

S60 devices come from the factory with some signal processing algorithms implemented using the public APIs (e.g. Photo Editor Application with image editing facilities, like sharpness improvements). There are also third-party signal processing applications as PhotoAcute [10] that implements a super-resolution algorithm based on multiple image correlation and interpolation that depends on the optical system of the device. In our implementation we employed a N70 mobile device and we used S60 SDK version 2nd FP3 without any capability restrictions [8]. We utilized OpenGL ES API to represent in 3D the unwrapped phase. Image manipulation is done in S60 with more APIs: one for image loading, one for image rotating, one for image converting, one for image scaling. We used them in order to load and manipulate information contained in interferogram image.

VI. EXPERIMENTAL RESULTS

We implemented the algorithms presented above in Matlab and, also, on a S60 platform. As we mentioned in the previous section, we have done an implementation on a real S60 device: Nokia N70 based on version 2 of S60 platform. The phase unwrapping algorithms we chose provided good conditions to test the S60 platform in a simple case (zigzag approach) and in a more complex one (the least-squares method).

First, we present some results of the above described algorithms obtained in Matlab. In this way, we have a clear image of what the algorithms should have as input and what they should output. In the second step, we show some screenshots of the Symbian S60 application and we offer specific implementation details.

The quality of the interferograms is affected, mainly, by a low signal-to-noise ratio, due to low reflection regions as those that contain sand or rivers.

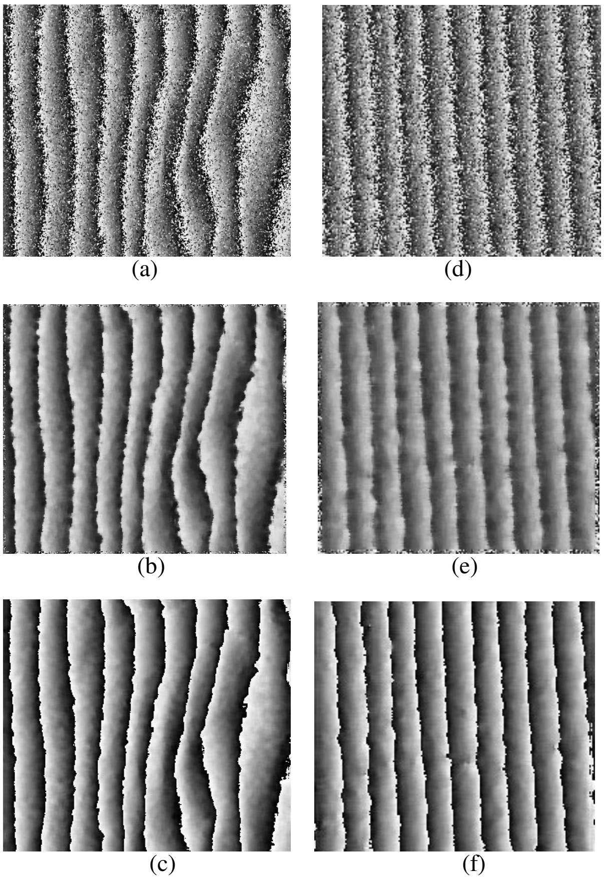

Figure 2. (a) Interferogram with noise; (b) Mean filtering for (a); (c) Vector filtering for (a); (d) Interferogram with more noise (lower SNR), and smaller distances between fringes; (e) Mean filtering for (d); (f) Vector filtering for (d).

In Figure 2, we show some results of the vector filtering when compared with mean filtering. The images show plain fields in China. Figure 2(a) shows an interferogram acquired with SIR-C system, while Figure 1(d) displays data provided by ERS 1/2 satellites tandem.

Vector filtering proved to be effective in noise removal, while keeping fringes' structure intact. Depending on the surface slope the fringes can be dense or rare. If the surface is a high variability, the interferogram will contain regions with dense fringes and, also, regions with rare fringes. In this case, if the SNR is low vector filtering can distort the fringe contours. Large filtering window deals well with rare fringes, while small filtering window deals well with dense fringes. Figures 3 and 4 present such case using and artificial interferogram.

There is no algorithm for phase unwrapping to produce good results if the interferogram image is not preprocessed against noise. In case the preprocessing is skipped, in the final result, there will be unwrapped regions for the local algorithms even for residue-cut ones, and, for the global algorithms, the final result, even if at the first sight it looks correct, it will contain errors (distortions). In our case, the zigzag phase unwrapping is a residua sensitive algorithm, and the least-squares approach can be considered to be a global one.

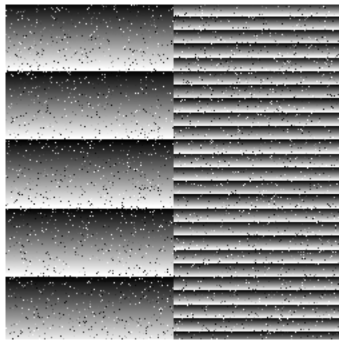

Figure 3. Interferogram with noise containing dense and rare fringes.

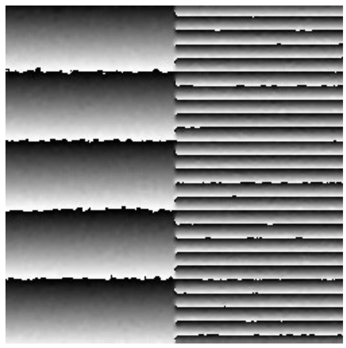

Figure 4. Vector filtering with a smaller window yields fringe distortions upon rare ones (for Figure 3).

The zigzag approach is affected by residua, by distortions that appear in the affected area and in the vicinity of it. The least-squares method tends to add uniformity to the surface in the affected regions, but it is also affected globally by the residua. Figure 5 presents results of the unwrapping algorithms using an artificial interferogram (an ideal mountain) with residua added.

Figure 6 shows a more complex artificial case: a noise affected interferogram that is unwrapped using vector filtering and the least-squares method. The result obtained is very good.

The phase unwrapping algorithms were also tested using real data acquired by the European satellites tandem ERS-1/ERS-2. In Figure 7, one can see a good quality interferogram of Mount Etna (Italy). The peak is approximately in the center of the image. The right side of the interferogram has a lower quality due to terrain reflectivity (volcanic rock).

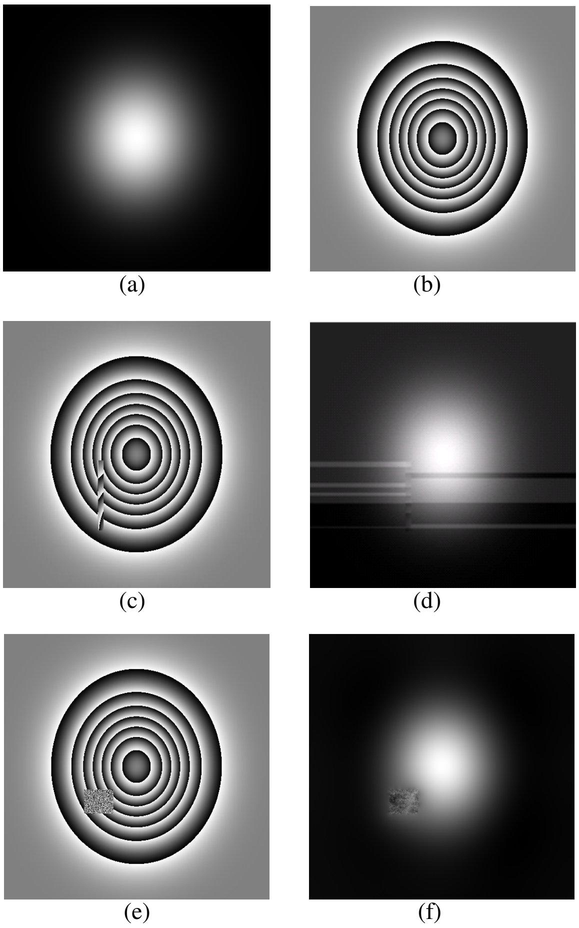

Figure 5. (a) Artificial elevation map image (it can be viewed as an ideal result of an unwrapping method); (b) The interferogram of it; (c) The interferogram having an added imperfection; (d) The elevation map (unwrapped phase) after using a residua sensitive algorithm (zig-zag approach); (e) The interferogram having an added square region with a lot of noise; (f) The elevation map (unwrapped phase) after using a global algorithm (least-squares approach).

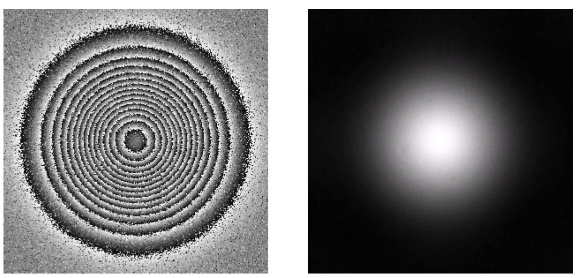

Figure 6. Noise affected artificial interferogram (left); Phase unwrapping result after using vector median filtering and the least-squares method (right).

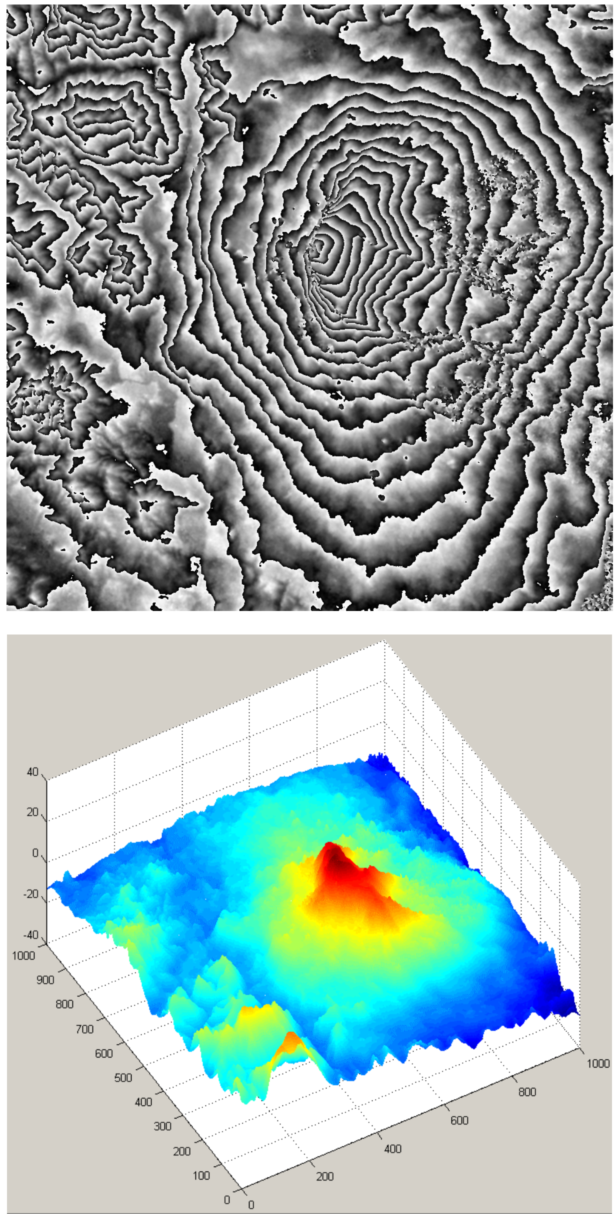

*Figure 7. Mount Etna interferogram (up)
Phase unwrapping result after using vector median filtering and the least-squares method – 3D view (down).*

For S60 platform the results were tested on emulator provided by S60 SDK publicly available, but, also, on a real device.

Figure 8 (a) presents the N70 device we used to implement the unwrapping algorithms. In Figure 8 (b), one can see the phone’s list with added applications and through them one can notice our application (called PhaseUnW). Figure 8 (c) reveals the options our S60 program offers. The user can load an interferogram image, can zoom in (phone’s keypad: 1) or zoom out (phone’s keypad: 3, scroll the image using navigation keys, and rotate the image with a 90 degrees step clockwise using selection key.

An example is presented in Figure 8 (d)-(e). The user

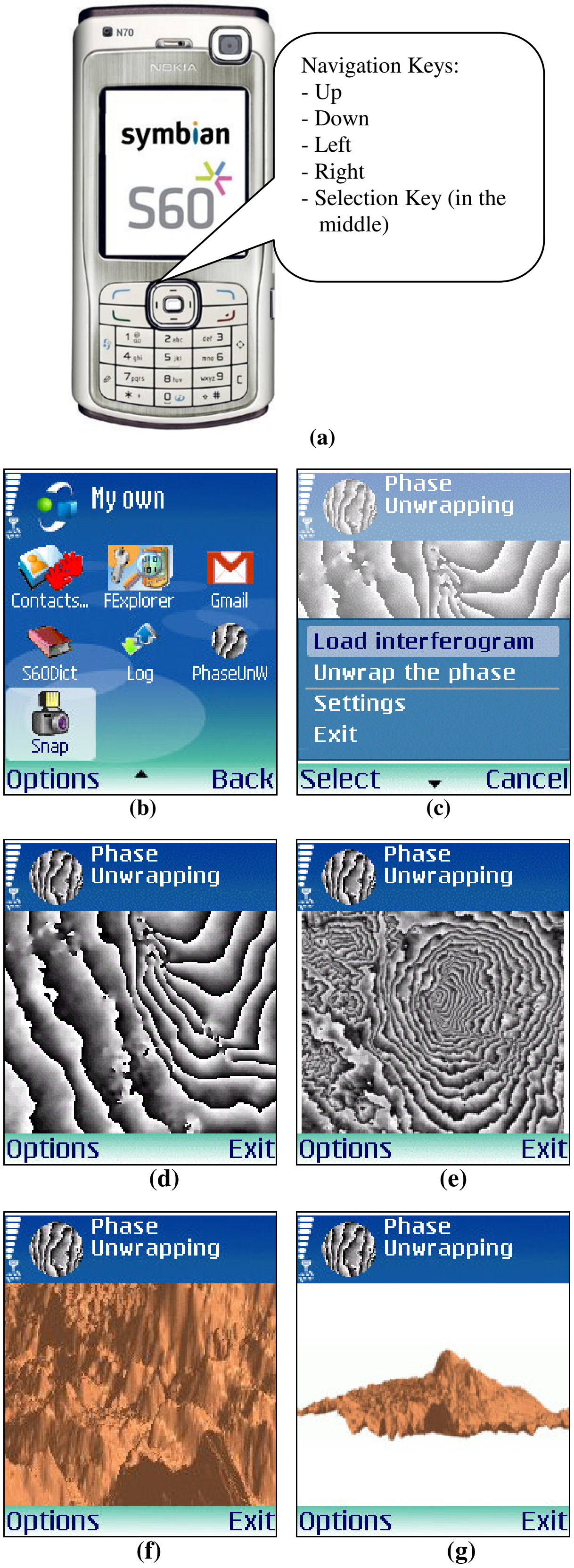

Figure 8. Symbian S60 application running on N70 mobile device.

can also make some settings: mean, vector or no filtering, and phase unwrapping using the zigzag algorithm or the least squares method. After loading the interferogram and making the settings, the user can select to unwrap the image containing the phase values. The result will be shown in a 3D representation – See Figure 8 (f)-(g).

There are options to zoom in and out the 3D view (phone's keypad: 1 and 3), to scroll the view using navigation keys, and to rotate in a 3D space the view (phone's keypad: 2, 4, 8, 6). In our implementation, the interferogram image can be loaded from the phone's built-in storage memory or from a memory supported by the phone.

Next, we compare the results obtained in Matlab (running on a PC with Intel Core 2 CPU 2GHZ, 2GB RAM) with the ones obtained on S60 device (N70 based on S60 ver 2). We acquired the same quality of the output. Both Matlab and S60 application outputted the phase unwrapping result also in the form of an uncompressed (to avoid distortions generated by compresion) TIFF grayscale image. In order to evaluate the computation time let us consider the inteferogram of Mount Etna (Figure 7) that has 1000x1000 pixels and vector median filter based noise removal. The computation time was about the same for the simple algorithm: zizag phase unwrapping (max. 1 sec). In case of the complex algorithm: the least-squares method, the S60 implementation needed about 2.30 min to complete, while Matlab processed it in 5 sec. A newer S60 device (e.g. ver 3) can get the results faster, since the platform is improved and the hardware is better, but we had no opportunity to test on such a device.

VII. CONCLUSIONS

In this paper we presented the implementation of two phase unwrapping algorithms (the zigzag approach and the least-squares method) on a Symbian S60 mobile device.

Phase unwrapping is a procedure that allows 3D representation for surfaces when the data is acquired using interferometry based techniques. An interferogram contains phase difference values wrapped onto a fixed range of angles 0-360 degrees. In order to compute surface heights and generate an elevation model, the interferogram have to be unwrapped.

We, also, implemented two noise removal algorithms: mean and vector filtering, for preprocessing the interferogram. The implementation was tested using SAR (Synthetic Aperture Radar) interferograms. Our S60 application performed well. However, it is hard to say that a specific algorithm architecture performs better. The results proved that multiple runs to analyze an image can increase the time non-linearly. For the tested algorithms, the device's memory can be problematic only if the input image is very large (i.e. larger than 5000x5000).

After doing this work, we can conclude that today's mobile devices offer enough processing and graphical power and development options, in order to be a good choice to test some new signal processing algorithms. Mobile devices become stronger and stronger and the development options are increasing. Android platform is a totally open and free option to develop for mobiles. It has just been launched by Google and its partners.

In our application, the data (i.e. the interferogram) is taken from the phone's memory, but this option can be developed further in order to get the data from a remote server using a 3G interface or a WLAN.

REFERENCES

[1] H.A. Zebker, and R.M. Goldstein, "Topographic Mapping From Interferometric Synthetic Aperture Radar Interferometry". *Journal of Geophysical Research*, 91(B5): 4993-4999, 1986.
[2] A.K. Gabriel, R.M. Goldstein., "Crossed Orbit Interferometry: Theory and Experimental Results From SIR-B", *International Journal of Remote Sensing,* 9(5): 857-872, 1988.
[3] F. Li, R.M.Goldstein," Studies of Multibaseline Spaceborne Interferometric Synthetic Aperture Radar", *IEEE Transactions on Geoscience and Remote Sensing*, 8: 88-97, 1990.
[4] H.A. Zebker, and Y. Lu, "Phase Unwrapping Algorithms for Radar Interferometry: Residue-Cut, Least-Squares, and Synthesis Algorithms", *J. Opt. Soc. Am. A*,Vol. 15, No. 3, March 1998.
[5] 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, 1994.
[6] B.R. Hunt, "Matrix Formulation of the Reconstruction of Phase Values from Phase Differences", *J. Opt. Soc. Am.* 69, 393–399 (1979).
[7] M.D. Pritt, "Phase Unwrapping by Means of Multigrid Techniques for Interferometric SAR", *IEEE Trans. Geosci. Remote Sens.* 34, 728–738, 1996.
[8] L. Edwards, R. Barker, *Staff of EMCC Software Ltd., Developing Series 60 Applications: A Guide for Symbian OS C++ Developers*, Addison-Wesley, 2004.
[9] Symbian S60 Platform, http://www.forum.nokia.com/.
[10] PhotoAcute — A SuperResolution Software implementing superresolution algorithm based on correlating several digital images and optical system-dependent interpolation for S60 platform, http://photoacute.com/mobile.